A

Practical Activity Report submitted for

Engineering Design Project-II (UTA024)

by

Garati	**102103188**
Iqman Singh Bhatia	**102103189**
Narinder Kaur	**102103190**
Jasleen Kaur	**102103191**
Jatinjot Singh	**102103192**

BE Second Year

Batch: 2CO7

Submitted to

Dr. Krishan

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

DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PATIALA, PUNJAB

JAN-JUNE 2023

ABSTRACT

The Engineering design Project is a one of its kind Project. In a curriculum of theoretical courses, this course provides us with an opportunity to implement the theoretical knowledge into real life applications. The application of Programming and Circuits in real time applications is what this course is all about. In this Course, we are supposed to construct an Arduino Uno based Buggy robot which has various sensors attached to it. The Arduino Uno circuit can be simulated using Breadboard, IR Sensors, Ultrasonic sensors and various other components. The buggy can be used in various modes. It can be operated as a line follower using IR sensors; it can detect obstacles using Ultrasonic sensors and become a remote-controlled device using Zigbee. Arduino programming is very similar to C and hence with light codes we can make this buggy work. This makes our work easier and also gives quicker results. Along with this, we also need to work on mechanics of buggy as well which is a new domain of knowledge for electronics and computers students. This project will definitely bring out the creativity of students and they will learn a lot about robotics.

DECLARATION

We hereby declare that the project work entitled **“Buggy”** for CS Lab department project report is carried out under the supervision of our instructor **Dr. Krishan** whose guidance helped us a lot in coming to the results of our reports. We assure that the conclusions drawn from this report are all our own research and are purely our efforts. We further assure you that all the things mentioned in this report is original without any plagiarism. This report has not been submitted to any university or institution except. Thapar Institute of Engineering and Technology, Patiala. We made this report to the best of our knowledge and carried out all the conclusions from the experiments all by ourselves.

TABLE OF CONTENTS

LIST OF EXPERIMENTS

Sr. No.	Experiment No.	Objective
1	**1**	Introduction to Arduino Microcontroller.
2	**2(a)**	Write a program in Arduino to blink a LED. (Single LED)
3	**2(b)**	Write a program in Arduino to blink a LED. (Multiple LED)
4	**3**	Write a program to design a pattern from sequence of multiple LED using for loop in Arduino.
5	**4**	Write a program to demonstrate sending data from the computer to the Arduino board and control brightness of LED.
6	**5**	Write a program for following pattern using for loop: ******************************* Roll_No.________________________________ ******************************* Name:__________________________________ ******************************* Branch:_________________________________ *******************************
7	**6(a)**	WAP for dimmer (Change in intensity of single LED bulb) using **digitalRead()**
8	**6(b)**	WAP for dimmer (Change in intensity of single LED bulb) using **analogRead()**
9	**7**	WAP to change the intensity of the given LEDs for the sequence **35214** in for both forward and reverse order.
10	**8**	WAP to demonstrate control of DC Motor using forward, backward, left, right turn motion and clock-wise/anti clock-wise rotation.
11	**9**	WAP to read values of IR Sensor using analog and digital read and convert buggy into normal line follower robocar.
12	**10**	WAP to demonstrate the use of ultrasonic sensor by integrating line follower robocar with obstacle avoidance capability.
13	**11(a)**	WAP to read the pulse width of gantry transmitter and trigger stop_buggy function by detecting individual gantry.
14	**11(b)**	WAP to demonstrate Xbee module communication between two PCs using X-CTU.
15	**12**	**Bronze Challenge:** Single buggy around track twice in clockwise direction, under full supervisory control. Buggy can detect an obstacle, Parks safely. Prints state of the track and buggy at each gantry stop.

LIST OF TABLES

LIST OF FIGURES

Experiment: 1

Objective: Introduction to Arduino Microcontroller.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Arduino Uno R3	NA

Table 1

Software Used: Arduino IDE v.1.8.19

Theory:

A microcontroller is a computer**.** Microcontrollers are "embedded" inside some other device so that they can control the features or actions of the product. A microcontroller also takes input from the device it is controlling and controls the device by sending signals to different components in the device. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on computer, used to write and upload computer code to the physical board.

Circuit diagram:

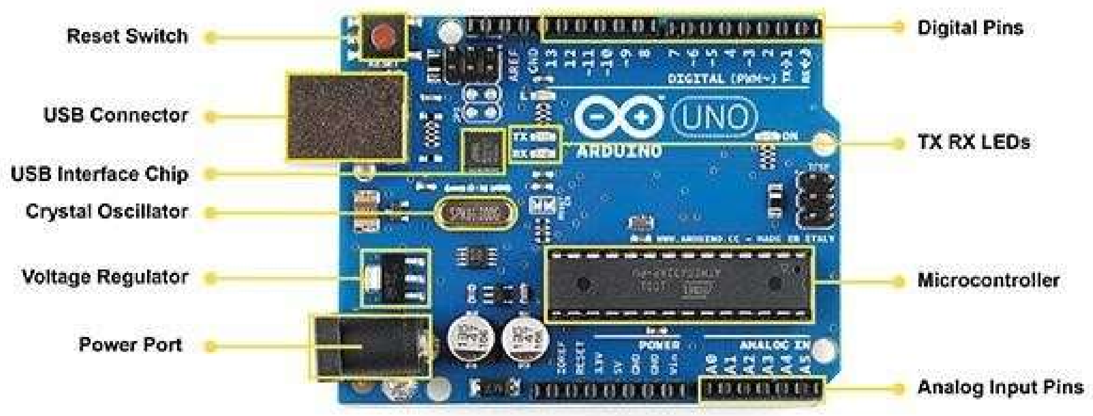


Fig 1 Arduino Uno R3

1. **Power Button:** Arduino board can be powered by using the USB cable from our computer. All we need to do is connect the USB cable to the USB connection.
2. **Power (Barrel Jack):** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

3. **Voltage Regulator:** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4. **Crystal Oscillator:** The crystal oscillator helps Arduino in dealing with time issues.
5. **Pins (3.3, 5, GND)**

   3.3V − Supply 3.3 output voltage

   5V − Supply 5 output voltage

   GND (Ground) − GND pins on the Arduino, any of which can be used to ground your circuit.
6. **Analog pins:**

   The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value which is readable.
7. **Digital I/O**

   The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic. The pins labelled "~" can be used to generate PWM.

Result Analysis:

In this experiment, we came across the concept of Arduino and the functionalities involved in it. We dealt with the concept of pins and basics of this Arduino Uno board, its different miniature analogies and how it basically works (surface approach).

Signature of Faculty Member

Experiment: 2(a)

Objective: Write a program in Arduino to blink a (Single) LED.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (1)	220 Ω
2.	Light Emitting Diode (1)	Red
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 2.1

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor**:
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

Circuit Connections:

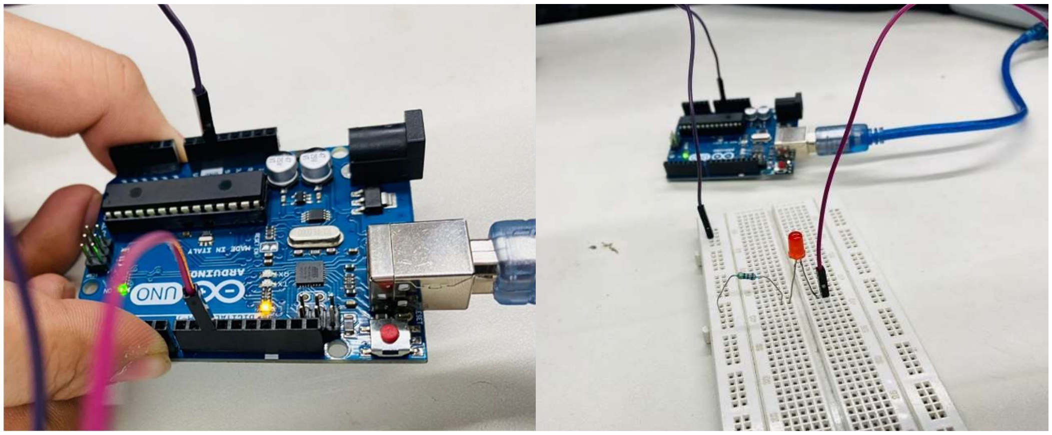


Fig 2.1 Connections for Exp 2(a)

Code:

```
void setup() {
  pinMode(10,OUTPUT);
}
void loop() {
  digitalWrite(10,HIGH);
  delay(500);
  digitalWrite(10,LOW);
   delay(500);
}
```

Result Analysis:

In this experiment, we have learnt how we can select the pin of an Arduino board and set it HIGH or LOW depending upon the usage. The delay time between the transition for the desired interval time could be set giving the command.

Signature of Faculty Member

Experiment: 2(b)

Objective: Write a program in Arduino to blink a LED. (Multiple)

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (3)	220 Ω
2.	Light Emitting Diode (3)	Red (2) /Yellow
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 2.2

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor:**
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

Logical Circuit Connections:

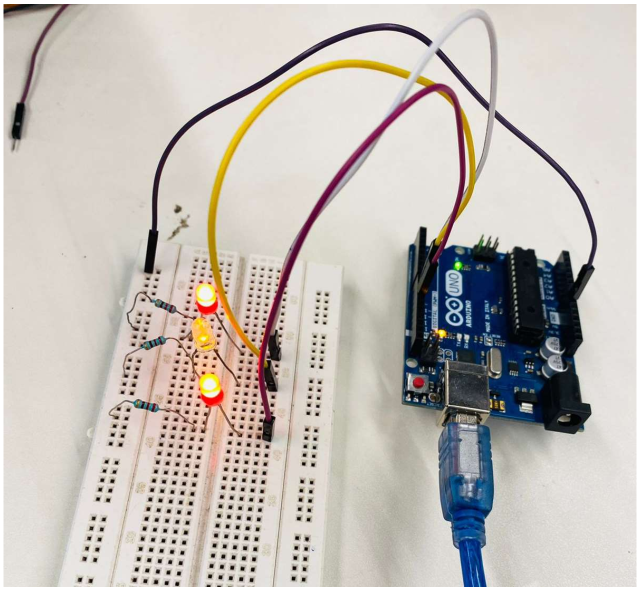


Fig 2.2 Connections for Exp 2(b)

Coding:

```
void setup() {
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}

void loop() {
  for(int i = 10;i>7;i--)
  {
    digitalWrite(i,HIGH);
    delay(500);
  }
  for(int i = 8;i<11;i++)
  {
    digitalWrite(i,LOW);
    delay(500);
  }
}
```

Result Analysis:

In this experiment, we have learnt how we can select the pin of an Arduino board and set it HIGH or LOW depending upon the usage. The delay time between the transition for the desired interval time could be set giving the command for not one but multiple pins at a time.

Signature of Faculty Member

Experiment: 3

Objective: Write a program to design a pattern (4-bit binary counter) from sequence of multiple LED using for loop in Arduino.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (4)	220 Ω
2.	Light Emitting Diode (LED) (4)	Red (3), Yellow
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 3

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor:**
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

Logical Circuit Connections:

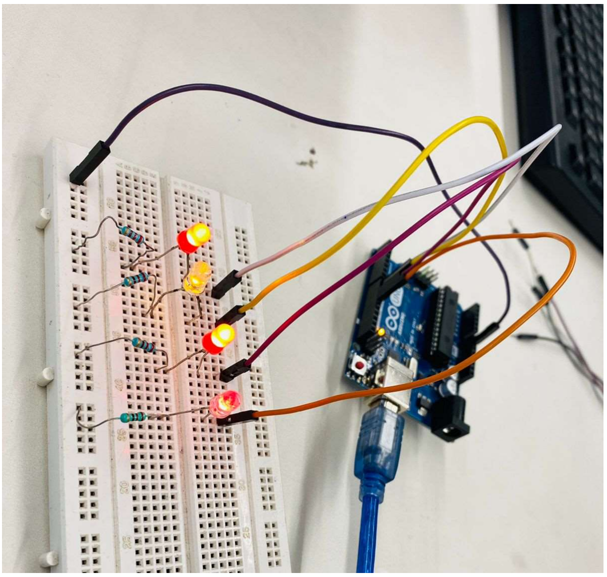


Fig 3 Connections for Exp 3

Coding:

```
void setup() {
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
}
void loop() {
  for(int i = 0;i<16;i++)
  {
    int a = i%2;
    int b = (i/2)%2;
    int c = (i/4)%2;
    int d = (i/8)%2;
   digitalWrite(11,a);
   digitalWrite(10,b);
   digitalWrite(9,c);
   digitalWrite(8,d);
   delay(1000);
  }
}
```

Result Analysis:

In this experiment, we have learnt how we can easily create a 4-bit binary counter. For loop could be employed in such symmetric cases to obtain desired results.

Signature of Faculty Member

Experiment: 4

Objective: Write a program to demonstrate sending data from the computer to the Arduino board and control brightness of LED.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (2)	220 Ω
2.	Light Emitting Diode (LED) (2)	Red
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 4

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor:**
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

Logical Circuit Connections:

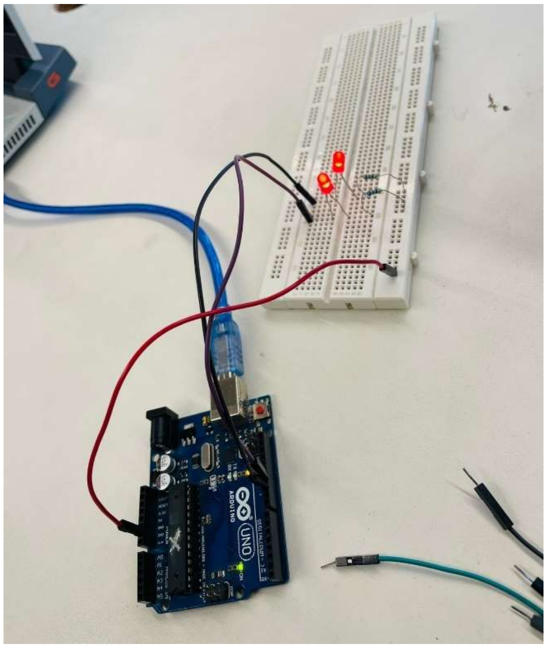


Fig 4 Connections for Exp 4

Coding:

```
void setup() {
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if(Serial.available()>0)
  {
    int val = Serial.parseInt();
    //Serial.print(val);
    analogWrite(10,val);
    analogWrite(11,val);
  }
}
```

Result Analysis:

In this experiment, we have learnt how we can control the input by sending it from the hardware. Here in case, the brightness of the LED was controlled by sending its value from the system using parseInt command.

Signature of Faculty Member

Experiment: 5

Objective: WAP for following pattern using for loop:

```
******************************
Roll No. ______________________________________
******************************
Name: ________________________________________
******************************
Branch:_______________________________________
******************************
```

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Arduino Uno R3 (1)	NA

Table 5

Software Used: Arduino IDE v.1.8.19

Theory:

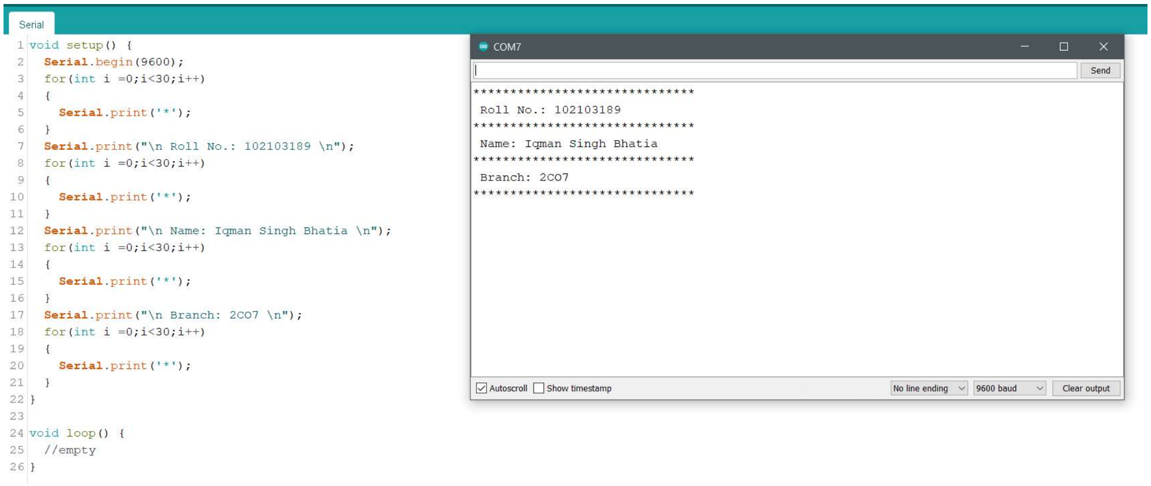
Arduino:

An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins that may be interacted to various circuits.

Coding:

```
void setup() {
  Serial.begin(9600);

  for(int i =0;i<30;i++)
  {    Serial.write('*');  }
  Serial.print("\n Roll No.: 102103189 \n");

  for(int i =0;i<30;i++)
  {    Serial.print('*');  }
  Serial.print("\n Name: Iqman Singh Bhatia \n");

  for(int i =0;i<30;i++)
  {    Serial.print('*');  }
  Serial.print("\n Branch: 2CO7 \n");

  for(int i =0;i<30;i++)
  {    Serial.print('*');  }
}
void loop() {
  //empty
}
```

Serial Monitor:



```
void setup() {
  Serial.begin(9600);
  for(int i =0;i<30;i++)
  {
    Serial.print('*');
  }
  Serial.print("\n Roll No.: 102103189 \n");
  for(int i =0;i<30;i++)
  {
    Serial.print('*');
  }
  Serial.print("\n Name: Iqman Singh Bhatia \n");
  for(int i =0;i<30;i++)
  {
    Serial.print('*');
  }
  Serial.print("\n Branch: 2CO7 \n");
  for(int i =0;i<30;i++)
  {
    Serial.print('*');
  }
}

void loop() {
  //empty
}
```

```
******************************
 Roll No.: 102103189 
******************************
 Name: Iqman Singh Bhatia 
******************************
 Branch: 2CO7 
******************************
```

Fig 5 Serial Monitor for Exp 5

Result Analysis:

In this experiment, we have learnt how to us serial communication commands in a programme and various usage of them in built up of a code to obtain certain set of desired results. ASCII values as whole were discussed in this experiment.

Signature of Faculty Member

Experiment: 6(a)

Objective: WAP for dimmer (Change in intensity of single LED bulb) using digitalRead()

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (4)	1k Ω
2.	Light Emitting Diode (LED) (1)	Yellow
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 6.1

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor:**
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

Logical Circuit Connections:

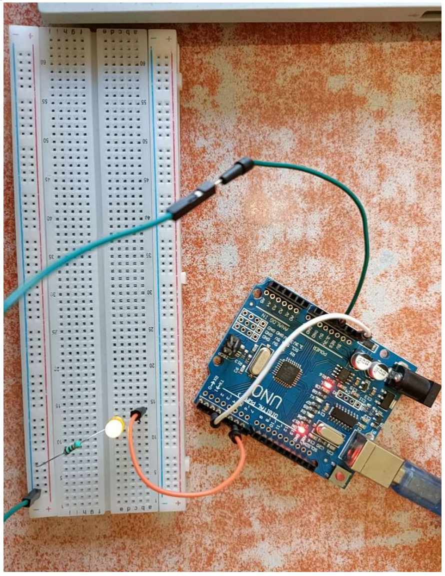


Fig 6.1.1 Pin 2 connected to 5V

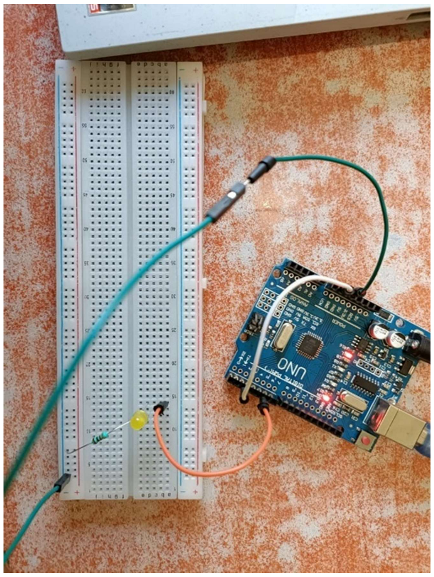


Fig 6.1.2 Pin 2 connected to GND

Coding:

```
void setup() {
  pinMode(5,OUTPUT);
  pinMode(2,INPUT);
}
void loop() {
  int val = digitalRead(2);
  if(val == HIGH){
    for(int j=0;j<256;j++){
      analogWrite(5,j);
      delay(10);
    }
    for(int j=255;j>=0;j--){
      analogWrite(5,j);
      delay(10);
    }
  }
  else
  {    analogWrite(5,0);}
  delay(1000);
}
```

Result Analysis:

We used analogWrite function to have a pattern of dimmer in LEDs. digitalRead function was used as a switch to start the dimming effect if input was HIGH. For loop in addition to analogWrite command is used for writing the program.

Signature of Faculty Member

Experiment: 6(b)

Objective: WAP for dimmer (Change in intensity of single LED bulb) using analogRead()

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (4)	1k Ω
2.	Light Emitting Diode (LED) (1)	Yellow
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 6.2

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor:**
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

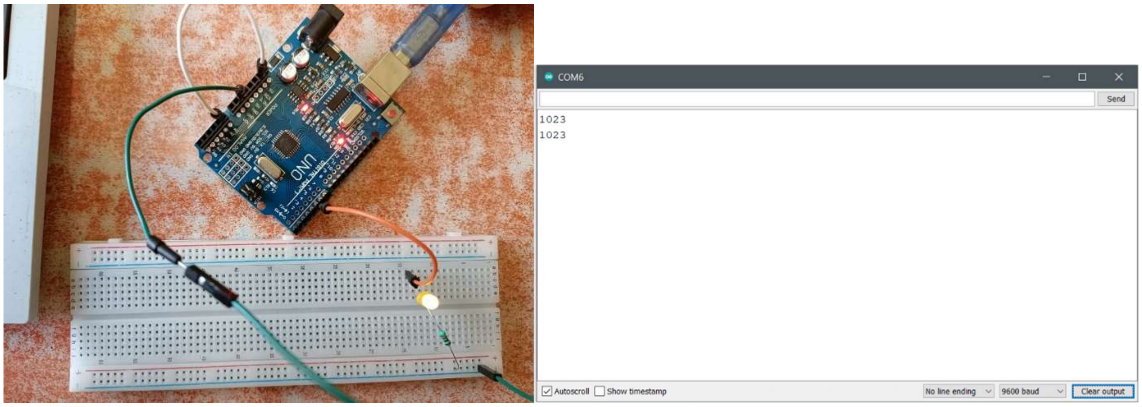
2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

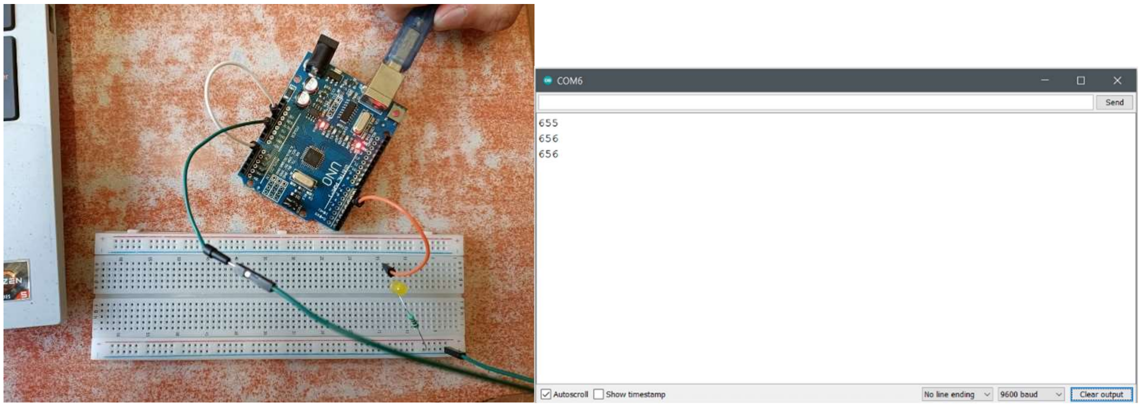
Logical Circuit Connections:



```
1023
1023
```

☑ Autoscroll ☐ Show timestamp No line ending 9600 baud

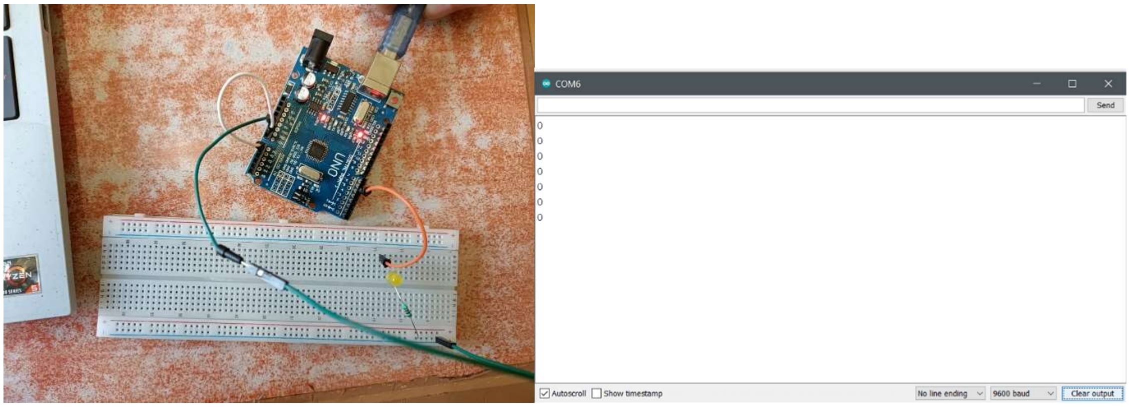
Fig 6.2.1 A0 Pin connected to 5V



```
655
656
656
```

☑ Autoscroll ☐ Show timestamp No line ending 9600 baud

Fig 6.2.2 A0 Pin connected to 3.3V



```
0
0
0
0
0
0
0
```

☑ Autoscroll ☐ Show timestamp No line ending 9600 baud

Fig 6.2.3 A0 Pin connected to GND

Coding:

```
void setup() {
  pinMode(5,OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  if(sensorValue >=1000)
  {
    for(int j=0;j<256;j++)
    {
      analogWrite(5,j);
      delay(10);
    }
    for(int j=255;j>=0;j--)
    {
      analogWrite(5,j);
      delay(10);
    }
  }
  else
  {
    analogWrite(5,0);
  }
      delay(1000);
}
```

Result Analysis:

We used analogWrite function to have a pattern of dimmer in LEDs. analogRead function was used as a switch to start the dimming effect if input was greater than 1000. For loop in addition to analogWrite command is used for writing the program.

Signature of Faculty Member

Experiment: 7

Objective: WAP to change the intensity of the given LEDs for the sequence 35214 in for both forward and reverse order.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Resistor (5)	1k Ω
2.	Light Emitting Diode (LED) (5)	Red (2), White (2), Yellow
3.	Arduino Uno R3 (1)	NA
4.	Breadboard (1)	NA
5.	Connecting Wires	NA

Table 7

Software Used: Arduino IDE v.1.8.19

Theory:

1. **Resistor:**
   A two terminal device in circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

2. **LED**:
   A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

3. **Arduino:**
   An Arduino Uno is a microcontroller board which is developed with multiple analog and digital input output pins used to create/test various circuits.

4. **Breadboard:**
   A breadboard is a rectangular plastic board with a bunch of tiny holes in it which let in to easily insert electronic components to prototype.

5. **Connecting Wires:**
   It provides a medium to an electrical current so that they can travel from one point on a circuit to another.

Logical Circuit Connections:

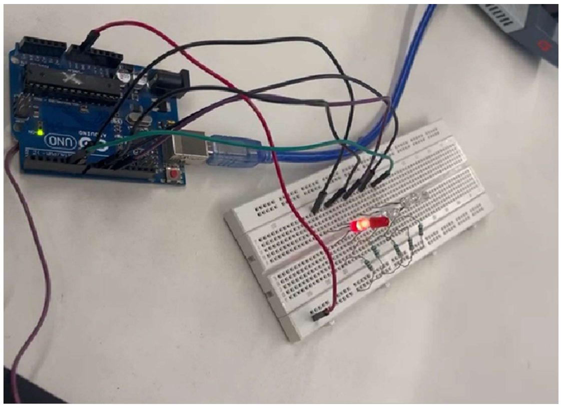


Fig 7 Connections for Exp 7

Coding:

```
// Serial Input: 35214
int n[5] = {5,6,9,10,11};
void setup() {
  pinMode(5,OUTPUT);   //led1
  pinMode(6,OUTPUT);   //led2
  pinMode(9,OUTPUT);   //led3
  pinMode(10,OUTPUT); //led4
  pinMode(11,OUTPUT); //led5
  Serial.begin(9600);
}
void loop() {
  if(Serial.available()>0)
  {
    int i = Serial.parseInt();
    for(int j=0;j<256;j++)
    { analogWrite(n[i-1],j);
      delay(10);    }
    for(int j=255;j>=0;j--)
    { analogWrite(n[i-1],j);
      delay(10);    }
    Serial.print(i);
  }
}
```

Result Analysis:

In this experiment, we have learnt how we can use serial communications and for loop to consider a particularly desired format for lighting LEDs in a pattern.

Signature of Faculty Member

Experiment: 8

Objective: WAP to demonstrate control of DC Motor using forward, backward, left, right turn motion and clock-wise/anti clock- wise rotation.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Arduino UNO R3	NA
2.	USB Cable	NA
3.	Nvis 3302ARD RoboCar	NA
4.	Connecting Wires	NA

Table 8

Software Used: Arduino IDE v.1.8.19

Theory:

The RoboCar is an assembled structure of Nvis 3302ARD. The machine is driven by DC motors which are powered by rechargeable batteries. A motor is a machine that converts electrical energy into mechanical energy(rotation). It's features includes – wireless control of robocar using XBee, detection of objects using ultrasonic sensor, detection of path using IR sensors and many more.

Model:

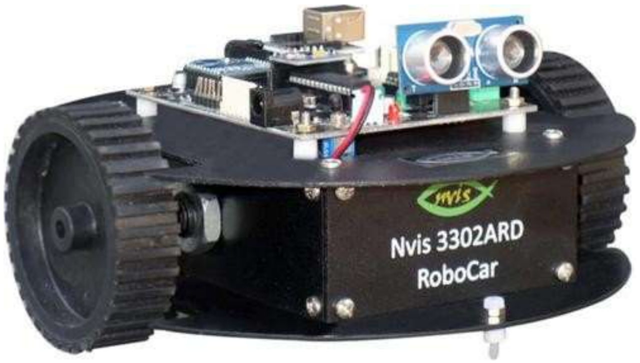


Fig 8 Nvis 3302 ARD RoboCar

Coding:

```
void setup() {
  pinMode(5,OUTPUT); //Right Tyre +ve
  pinMode(6,OUTPUT); //Right Tyre -ve
  pinMode(7,OUTPUT); //Left Tyre +ve
  pinMode(8,OUTPUT); //Left Tyre -ve
}
```

```
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 Left()
{
  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
}
void Right()
{
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
}
void Anti()
{
  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(8,LOW);
}
void Clock()
{
  digitalWrite(5,LOW);
  digitalWrite(6,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
}

void loop() {
  Forward();
  delay(3000);
  Left();
  delay(1250);
  Forward();
  delay(3000);
  Left();
  delay(1250);
  Forward();
  delay(3000);
  Left();
  delay(1250);
  Forward();
  delay(3000);
  Left();
  delay(1250);
  Right();
  delay(1500);
  Backward();
  delay(3000);
}
```

Result Analysis:

In this experiment, we have learnt how to use a DC Motor for the forward, backward, left, right turn motion and clock-wise/anti clock-wise rotation in a RoboCar.

Signature of Faculty Member

Experiment: 9

Objective: WAP to read values of IR Sensor using analog and digital read and convert buggy into normal line follower robocar.

Hardware Component Used:

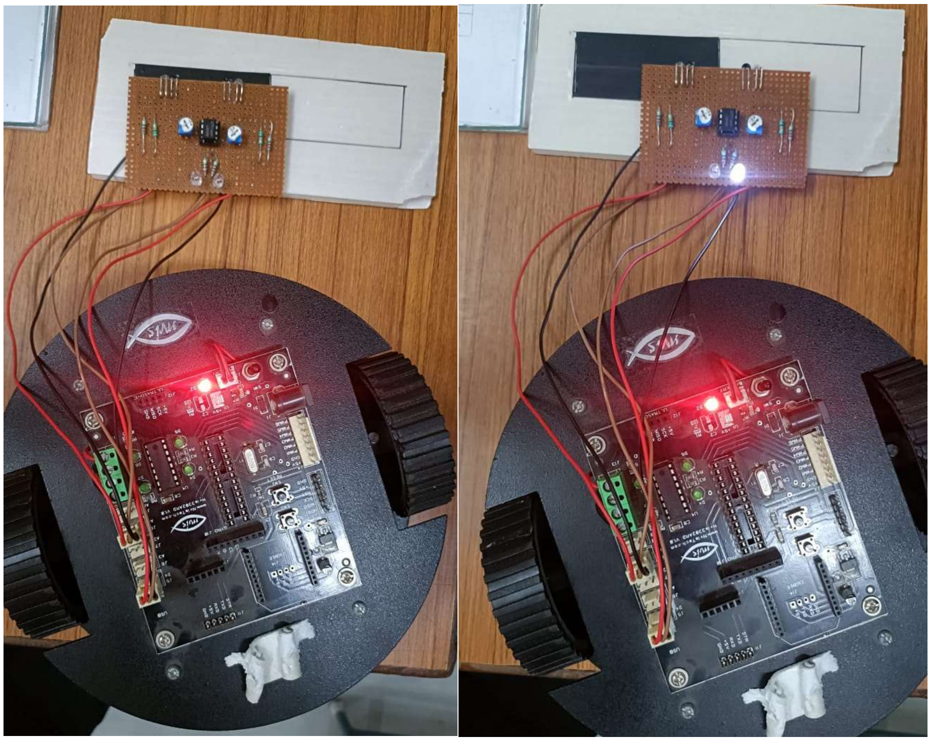
Sr. No	Name of Components	Value
1.	Arduino UNO R3	NA
2.	USB Cable	NA
3.	Nvis 3302ARD RoboCar	NA
4.	IR Sensor Pairs (2)	NA
5.	Connecting Wires	NA

Software Used: Arduino IDE v.1.8.19

Theory:

IR sensor basically works on intensity of light RGB code for white is (255,255,255) and for Black is (0,0,0). An IR sensor is a device that emits signals in order to sense some aspects of the surroundings which detects IR radiation falling on it. The emitter is an IR LED (Light Emitting Diode) and the detector is an IR photodiode which is sensitive to IR light. In our buggy, the IR sensor helps it to move only on the black lines of our path defined. We use the predefined functions from the previous experiment to control the movement of the buggy on the path in the Arduino IDE which in turn gives the instructions to the IR sensor according to values read by the IR sensor on the analog pins.

Model:



Coding:

```
void setup() {
  pinMode(5,OUTPUT); //Right Tyre +ve
  pinMode(6,OUTPUT); //Right Tyre -ve
  pinMode(7,OUTPUT); //Left Tyre +ve
  pinMode(8,OUTPUT); //Left Tyre -ve
  pinMode(A0,INPUT);
  pinMode(A1,INPUT); }

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 Left()
{  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
}
void Right()
{  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
}
void Anti()
{  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(8,LOW);
}
void Clock()
{  digitalWrite(5,LOW);
  digitalWrite(6,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
}
void Stop()
{  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
}
void loop() {
  Forward();
  int d0 = digitalRead(A0);
  int d1 = digitalRead(A1);
  int a0 = analogRead(A0);
  int a1 = analogRead(A1);
  if(d0 == 0 && d1 == 0)
  {    Forward();
    delay(100);    }
  else if(d0 ==0)
  {    Right();       }
  else if(d1 == 0)
  {    Left();        }
}
```

Result Analysis:

In this Experiment we learnt how to take input from IR Sensors using analog pins and also that how can we make the Buggy follow the black path and execute the proper path follow of the buggy by adjusting the sensitivity of the IR Sensor.

Signature of Faculty Member

Experiment: 10

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

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Arduino UNO R3	NA
2.	USB Cable	NA
3.	Nvis 3302ARD RoboCar	NA
4.	IR Sensor Pairs (2)	NA
5.	Ultrasonic Sensor HC-SR04	NA
6.	Connecting Wires	NA

Software Used: Arduino IDE v.1.8.19

Theory:

An ultrasonic sensor is an instrument that measures the distance to an object and detects the obstacle using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. Thus, we use the ultrasonic sensor on the buggy to measure the distance between the buggy and obstacle. We stop the buggy as soon as the distance between the buggy and the obstacle is less than 15cm. Hence, we design the code to control the working of the ultrasonic sensor on the robocar. The HC-SR04 module hosts the ultrasonic transmitter, the receiver and control circuit. The HC-SR04 has four pins namely Vcc, Trigger, Echo, GND.

Model:

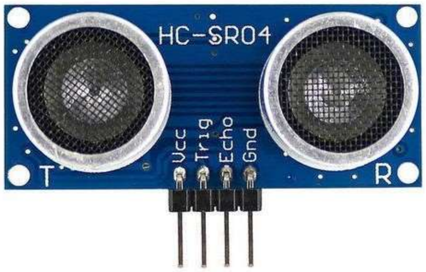


Fig 10 Ultrasonic Sensor

Coding:

```
#include<NewPing.h>
const int trigPin = 13;
const int echoPin = 12;
const int maxDistance  = 20;
NewPing sonar(trigPin,echoPin,maxDistance);
int distanceCm;
```

```
void setup() {
 pinMode(5,OUTPUT); //Right Tyre +ve
 pinMode(6,OUTPUT); //Right Tyre -ve
 pinMode(7,OUTPUT); //Left Tyre +ve
 pinMode(8,OUTPUT); //Left Tyre -ve
 pinMode(A0,INPUT);
 pinMode(A1,INPUT);
 pinMode(trigPin,OUTPUT);
 pinMode(echoPin,INPUT);
 Serial.begin(9600);
}

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 Left()
{  digitalWrite(5,HIGH);
 digitalWrite(6,LOW);
 digitalWrite(7,LOW);
 digitalWrite(8,LOW);
}
void Right()
{  digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 digitalWrite(7,LOW);
 digitalWrite(8,HIGH);
}
void Anti()
{  digitalWrite(5,HIGH);
 digitalWrite(6,LOW);
 digitalWrite(7,HIGH);
 digitalWrite(8,LOW);
}
void Clock()
{  digitalWrite(5,LOW);
 digitalWrite(6,HIGH);
 digitalWrite(7,LOW);
 digitalWrite(8,HIGH);
}
void Stop()
{  digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 digitalWrite(7,LOW);
 digitalWrite(8,LOW);
}

void loop() {
 Forward();
 int d0 = digitalRead(A0);
 int d1 = digitalRead(A1);
 if(d0 == 0 && d1 == 0)
 {   Forward();
    delay(150);
 }
 else if(d0 ==0)
 {    Right();  }
 else if(d1 == 0)
 {    Left();    }
 distanceCm = sonar.ping_cm();
 Serial.println(distanceCm);
 if(distanceCm>0 && distanceCm<15 )
 {    Stop();
      delay(500);  }
}
```

Result Analysis:

In this experiment, we learnt how to use the Ultrasonic sensor on the buggy for distance measurement. We also learned to design the code for controlling the motion of the buggy and detecting the obstacles in its path.

Signature of Faculty Member

Experiment: 11

Objective:

a) WAP to read the pulse width of gantry transmitter and trigger stop_buggy function by detecting individual gantry.

b) WAP to demonstrate Xbee module communication between two PCs using X-CTU.

Hardware Component Used:

Sr. No	Name of Components	Value
1.	Arduino UNO R3	NA
2.	USB Cable	NA
3.	Nvis 3302ARD RoboCar	NA
4.	IR Sensor Pairs (2)	NA
5.	Ultrasonic Sensor HC-SR04	NA
6.	XBee Module	NA
7.	Reciever Circuit	NA

Software Used: Arduino IDE v.1.8.19

Theory:

XBee module is used for the communication between 2 PC's using the X- CTU software. It is configured first and then attached to the buggy so that it can be controlled from the X-CTU software on its own. As soon as the receiver on the top of buggy receives signals from the transmitter that is attached on the gantry, our buggy stops.

Model:

Fig 11 XBee Module

Coding:

```
#include<NewPing.h>
bool start = false;
const int trigPin = 13;
const int echoPin = 12;
const int maxDistance  = 20;
NewPing sonar(trigPin,echoPin,maxDistance);
int distanceCm;
int prevGantry = 0;
```

```
void setup() {
  pinMode(5,OUTPUT); //Right Tyre +ve
  pinMode(6,OUTPUT); //Right Tyre -ve
  pinMode(7,OUTPUT); //Left Tyre +ve
  pinMode(8,OUTPUT); //Left Tyre -ve
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT); // Reciever
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
  Serial.begin(9600);
}

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 Left()
{  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);}
void Right()
{  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);}
void Stop()
{  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);}
void loop() {
  if(Serial.available())
  {
    char a = Serial.read();
    else if(a=='x')
    {
      start =true;
    }
  }
    if(start)
    {
      int d0 = digitalRead(A0);
      int d1 = digitalRead(A1);
      int g = pulseIn(A2,HIGH,5000);
      if(g>500     &&     g<1500     &&
prevGantry!=1)
      {
        Serial.println("gantry 1");
        Stop();
        delay(1000);
        prevGantry = 1;
      }
      else   if(g>1500   &&   g<2500   &&
prevGantry!=2)
      {
        Serial.println("gantry 2");
        Stop();
        delay(1000);
        prevGantry = 2;
      }
      else   if(g>2500   &&   g<3500   &&
prevGantry!=3)
      {
        Serial.println("gantry 3");
        Stop();
        delay(1000);
        prevGantry = 3;
      }

      if(d0 == 0 && d1 == 0)
      {  Forward();
          delay(150);    }
      else if(d0 == 0)
      {   Left();          }
      else if(d1 == 0)
      {   Right();       }
      else
      {   Forward();    }
    }
    distanceCm = sonar.ping_cm();
    if(distanceCm>0 && distanceCm<15 )
    { Stop();
      delay(500);   }
}
```

Result Analysis:

In this experiment we learnt about XBee Module and the X-CTU application used to configure the XBee. And also to stop the buggy at the gantry and display the distance respectively.

Signature of Faculty Member

Experiment: 12 (Bronze Challenge)

Objective:

Single buggy around track twice in clockwise direction, under full supervisory control. Buggy can detect an obstacle, Parks safely. Prints state of the track and buggy at each gantry stop.

Hardware Component Used:

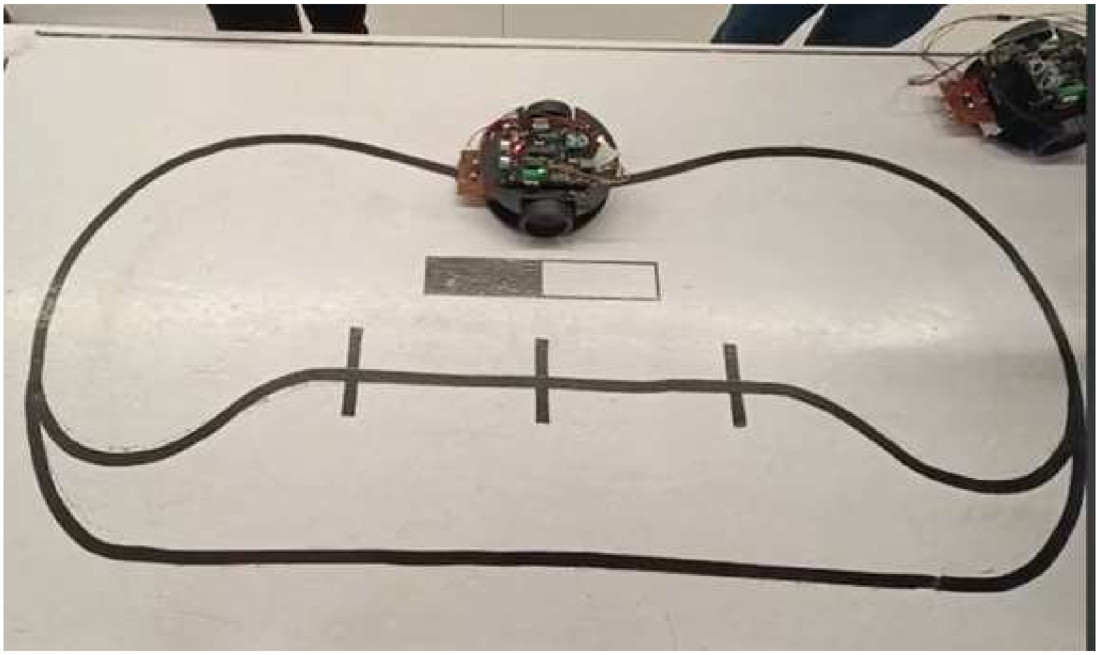
Sr. No	Name of Components	Value
1.	Arduino UNO R3	NA
2.	USB Cable	NA
3.	Nvis 3302ARD RoboCar	NA
4.	IR Sensor Pairs (2)	NA
5.	Ultrasonic Sensor HC-SR04	NA
6.	XBee Module	NA
7.	Reciever Circuit	NA

Software Used: Arduino IDE v.1.8.19

Theory:

The bronze challenge i.e. the buggy is required to traverse the whole path twice in clockwise direction while following the path, detect obstacles, park at the right position, stop at each of the three gantry and work with XBee configuration to give initial start to the buggy. This is done with the help of all the circuits prepared earlier and all the experiments performed. We are using the transmitter, receiver and the IR circuits soldered on our buggy to achieve this challenge.

Model:



Coding:

```
#include<NewPing.h>
bool start = false;     //init start off
const int trigPin = 13;
const int echoPin = 12;
const int maxDistance  = 20;
NewPing sonar(trigPin,echoPin,maxDistance);
int distanceCm;
int counter = 1;
int prevGantry = 0;
unsigned long prevTime = millis();
unsigned long currentTime;

void setup() {
  pinMode(5,OUTPUT); //Right Tyre +ve
  pinMode(6,OUTPUT); //Right Tyre -ve
  pinMode(7,OUTPUT); //Left Tyre +ve
  pinMode(8,OUTPUT); //Left Tyre -ve
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT); // Reciever
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
  Serial.begin(9600);
}
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 Left()
{  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
}
void Right()
{  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
}
void Anti()
{  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(8,LOW);
}
void Clock()
{  digitalWrite(5,LOW);
  digitalWrite(6,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
}
void Stop()
{  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
}
void loop() {
  if(Serial.available())
  {
    char a = Serial.read();
    if(a=='x')
    {
      start =true;
    }
  }
    if(start)
    {
      int d0 = digitalRead(A0);
      int d1 = digitalRead(A1);
      int g = pulseIn(A2,HIGH,5000);
      if(g>500 && g<1500 &&
prevGantry!=1)
      {
        Serial.println("gantry 1");
        Stop();
        delay(1000);
    prevGantry = 1;
   }
   else if(g>1500 && g<2500 &&
prevGantry!=2)
   {
    Serial.println("gantry 2");
    Stop();
    delay(1000);
    prevGantry = 2;
   }
   else if(g>2500 && g<3500 &&
prevGantry!=3)
   {
    Serial.println("gantry 3");
     delay(25);
    }
    else if(currentTime-prevTime>750
&& counter == 1)
    {
     prevTime = millis();
     Left();
     delay(20);
    }
    else if(currentTime-prevTime>1000
&& counter == 2)
    {
     //out
     prevTime = millis();
     Forward();
     delay(13);
    }
    else if(currentTime-prevTime>1000
&& counter == 3)
    {
     //in
     prevTime = millis();
     // counter 4
     Left();
     delay(100);
    }
    else if(currentTime-prevTime>500
&& counter == 4)
    {
     //parking 1
     prevTime = millis();

}
```

```
    Stop();
    delay(1000);
    prevGantry = 3;
   }
   if(d0 == 0 && d1 == 0)
   {
    currentTime = millis();
    if(currentTime-prevTime>500 &&
counter == 0)
    {
     //Parking Line
     prevTime = millis();
     //counter 1
     Forward();
     Forward();
     delay(10);
    }
    else if (currentTime-prevTime>500
&& counter >= 5)
    {
     Serial.println("Parked");
     Stop();
     start = false;
    }
    Serial.print("BB Counter: ");
    Serial.println(counter);
   }
   else if(d0 == 0)
   {
    Left();
   }
   else if(d1 == 0)
   {
    Right();
   }
   else
   {
    Forward();
   }
   }
  distanceCm = sonar.ping_cm();
  if(distanceCm>0 && distanceCm<15 )
  {
   Stop();
   delay(500);
  }
```

Result Analysis:

In this experiment we are performing the bronze challenge i.e. the buggy is required to traverse the whole path twice, detect obstacles, park at the right position, stop at each gantry and work with XBee configuration. We understood the working of the working of the buggy along with the working of various sensors and XBee for wireless communication.

Signature of Faculty Member