B M S COLLEGE OF ENGINEERING

(An Autonomous Institution Affiliated to VTU, Belagavi)

Post Box No.: 1908, Bull Temple Road, Bengaluru – 560 019

DEPARTMENT OF MACHINE LEARNING

Academic Year: 2022-2023 (Session: Oct 2022 - Feb 2023)

LOGIC DESIGN AND COMPUTER ARCHITECTURE(22AM3ESLDA)

ALTERNATIVE ASSESSMENT TOOL (AAT)

LASER COMMUNICATION BY DECIMAL TO BCD CONVERSION

Submitted by

Student Name:	Adithya G Jayanth, Amal Srivastava
USN:	1BM21AI008, 1BM21AI013
Date:	09/01/2023
Semester & Section:	III semester, 3A
Total Pages:	15
Student Signature:	

Valuation Report (to be filled by the faculty)

Score:	
Comments:	
Faculty In-charge:	Dr. Seemanthini K
Faculty Signature: with date	

1. INTRODUCTION

Binary Number System is a type of Number Representation technique. It is most popularly used in digital systems. Binary system is used for representing binary quantities which can be represented by any device that has only two operating states or possible conditions. For example, a switch has only two states: open or close. In the Binary System, there are only two symbols or possible digit values, i.e., 0 and 1. This can be represented by any device that has only 2 operating states or possible conditions.

Our modern world cannot be imagined without the avid use of the binary number system. Thus, it becomes necessary to understand the system and its conversion to and from the decimal system.

This project demonstrates the use of binary coded decimal to decimal conversion and vice-versa in a simple point to point LASER communication system. This is a hardware implementation using Arduino. This also demonstrates its interfacing with a 7-segment display.

Parts required:

1. Arduino board along with USB B cable (x2)

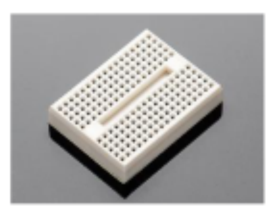
The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment implemented in embedded C), via a type B USB cable.

2. LASER diode

A laser diode (LD, also injection laser diode or ILD, or diode laser) is a semiconductor device similar to a light-emitting diode in which a diode pumped directly with electrical current can create lasing conditions at the diode's junction. This will be used to transmit the bits.

3. Bread Board

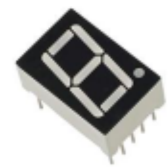
A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. They are useful as circuits can be implemented without soldering.



4. LEDs (x2)

To indicate the status of the received bits.

5. 7-segment display



Seven segment displays are the output display device that provides a way to display information in the form of images or text or decimal numbers. The received number will be displayed on a 7-segment display.

6. Light Dependent Resistor (LDR)

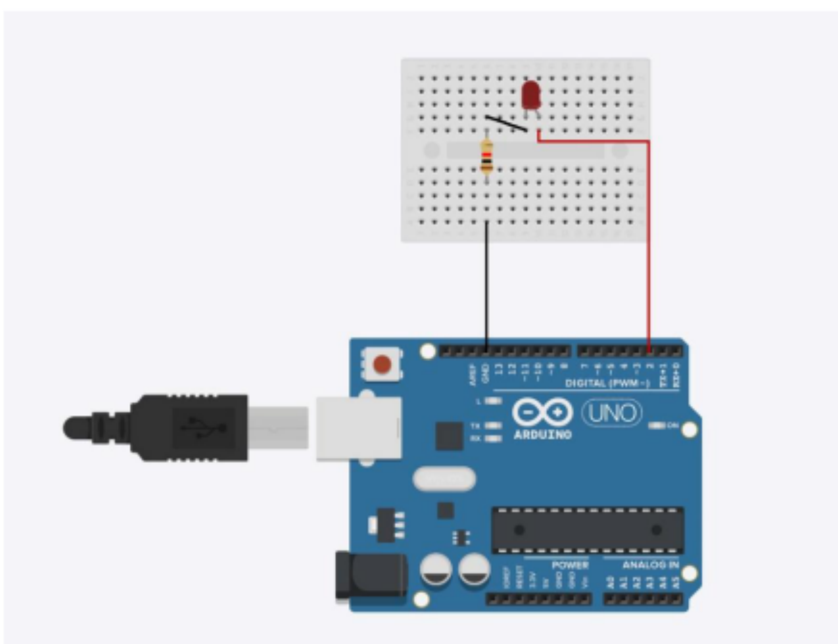
A photoresistor (also known as a photocell, or light-dependent resistor, LDR, or photo-conductive cell) is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface. This will be used to detect the LASER pulses.

7. Jumper cables

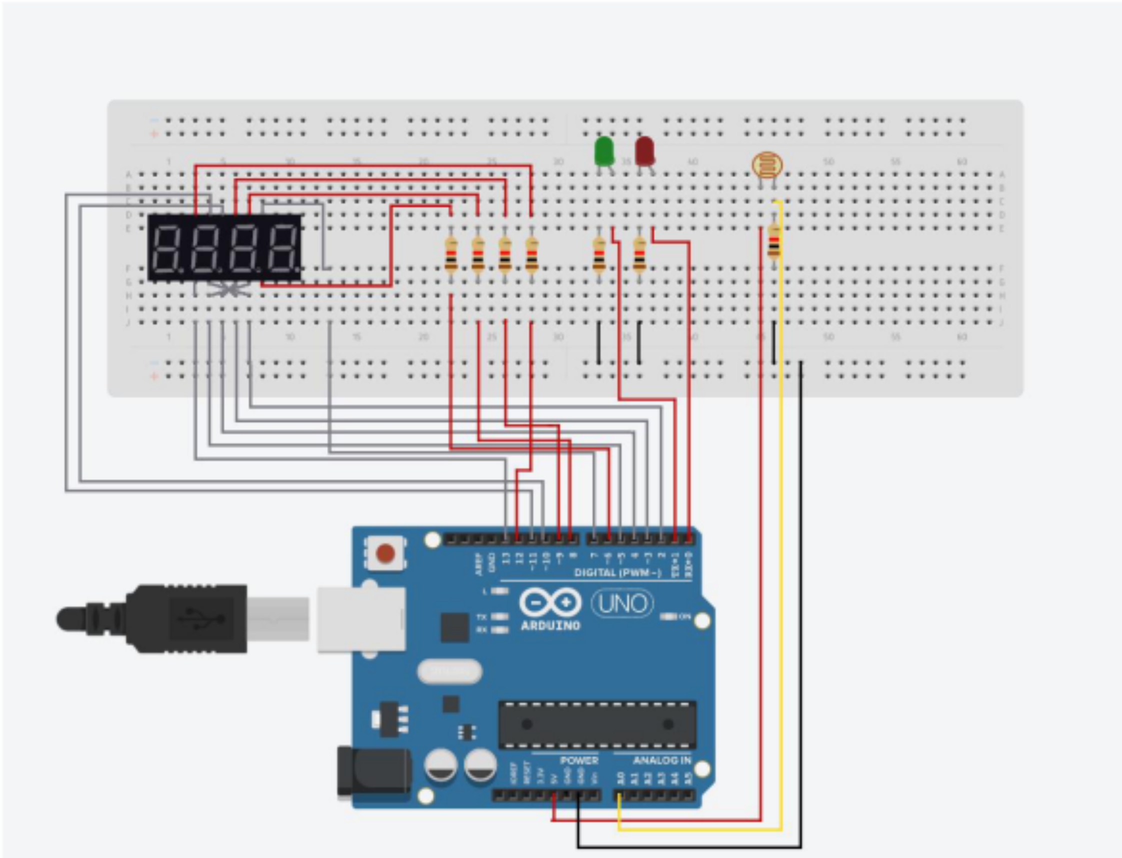
These are wires that will be used to connect components on the breadboard.

Circuit Connections:

Transmitter side:



Receiver's side:



Design:

In this laser communication system, a laser beam is used as a carrier. The LASER used is a red semiconductor LASER with a wavelength of 650nm. A wireless laser link (through a laser diode) is used to transmit information from one end to another in its line of sight. The sender takes the number in decimal and converts it into a binary coded decimal of 4 bits. The bits are then made into a 10 bit frame, starting with the start and stop bits as 111 to identify the frame. Then it is transmitted from the most significant bit to least significant. The laser transmits a one second pulse for high and no stimulus for one second for low.

Receiver will be taking input using a Light Dependent Resistor. The receiver maintains a 10-bit binary array with every element initialised to 0. Every 1 second, the bits are left-shifted by 1-bit and the MSB is discarded. The new LSB is read from the LDR. When the LDR receives the LASER light, the signal is interpreted as a high signal and when no light is received, it is interpreted as a low signal. Upon a high signal, 1 led is set to high to indicate the occurrence of a single bit of data. The 10-bit frame is continuously checked to see if a frame is detected. If the correct start and stop bits are identified, the total frame received is converted back to decimal

and displayed on a 7-segment display. A second led is lit up for 8 seconds along with the display to indicate the detection of a frame.

The project is built using arduino boards and implemented in embedded C. Both the receiver and transmitter require the use of 1 arduino board. The transmitter needs a computer system with the Arduino IDE installed to input data to the LASER.
The receiver circuit is standalone and displays the data on a 7-segment display

2. Embedded C Code

Transmitter's code:

```
//code is written in the arduino IDE software.
int laser = 2;

void setup() {
  pinMode(laser,OUTPUT);
  int num;
  Serial.begin(9600);
}
void loop() {
  if(Serial.available() > 0)  {
        int incomingData= Serial.read();
        if((int)incomingData-48 > -1){
        send_data((int)incomingData-48);
        Serial.println((int)incomingData -48);}
  }
}

void send_data(int n){
  byte y = n;
  digitalWrite(laser,HIGH);
  delay(3000);
  digitalWrite(laser,LOW);
  for(int i=3; i>=0; i--){
        bool m = bitRead(y, i);
        if(m == 1){
        digitalWrite(laser,HIGH);
        }
        delay(1000);
        digitalWrite(laser,LOW);
  }
  digitalWrite(laser,HIGH);
  delay(3000);
  digitalWrite(laser,LOW);
}
```

Receiver's code:

```
//NOTE : here the implementation has been done using a 4 digit 7-segment display
//with only one active digit.

int ldr;
int led1 = 0;
int led2 = 1;
int num = -1;
int pinA = 11;
int pinB = 7;
int pinC = 4;
int pinD = 2;
int pinE = 13;
int pinF = 10;
int pinG = 5;
int pinDP = 3;
int D1 = 12;
int D2 = 9;
int D3 = 8;
int D4 = 6;
int a[10];

void setup() {

  pinMode(A0,INPUT);
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(pinC, OUTPUT);
  pinMode(pinD, OUTPUT);
  pinMode(pinE, OUTPUT);
  pinMode(pinF, OUTPUT);
  pinMode(pinG, OUTPUT);
  pinMode(pinDP, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);

  for(int i= 0;i<10;i++) a[i] = 0;
}

void loop() {

  digitalWrite(led2, LOW);
  digitalWrite(led1, LOW);
  ldr = analogRead(A0);
```

```
    for(int i=9;i>0;i--){
            a[i] = a[i-1];
    }
    if(ldr>60){
            a[0]=1;
            digitalWrite(led1,HIGH);

    }
    else{
            a[0]=0;
            digitalWrite(led1,LOW);

    }
    if(((a[9]==1)&&(a[8]==1)&&(a[7]==1))                                    &&
  ((a[0]==1)&&(a[1]==1)&&(a[2]==1))){
            num = a[3]+2*a[4]+4*a[5]+8*a[6];

            switch(num){
            case 1:
                    digit3();one();
                    digitalWrite(led1, LOW);
                    digitalWrite(led2, HIGH);
                    delay(8000);
                    turnOffAllSegments();
                    digitalWrite(led2, LOW);
                    break;
            case 2:
                    digit3();two();
                    digitalWrite(led1, LOW);
                    digitalWrite(led2, HIGH);
                    delay(8000);
                    turnOffAllSegments();
                    digitalWrite(led2, LOW);
                    break;
            case 3:
                    digit3();three();
                    digitalWrite(led1, LOW);
                    digitalWrite(led2, HIGH);
                    delay(8000);
                    turnOffAllSegments();
                    digitalWrite(led2, LOW);
                    break;
            case 4:
                    digit3();four();
                    digitalWrite(led1, LOW);
                    digitalWrite(led2, HIGH);
                    delay(8000);
                    turnOffAllSegments();
                    digitalWrite(led2, LOW);
                    break;
```

```
case 5:
        digit3();five();
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        delay(8000);
        turnOffAllSegments();
        digitalWrite(led2, LOW);
        break;
case 6:
        digit3();six();
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        delay(8000);
        turnOffAllSegments();
        digitalWrite(led2, LOW);
        break;
case 7:
        digit3();seven();
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        delay(8000);
        turnOffAllSegments();
        digitalWrite(led2, LOW);
        break;
case 8:
        digit3();eight();
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        delay(8000);
        turnOffAllSegments();
        digitalWrite(led2, LOW);
        break;
case 9:
        digit3();nine();
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        delay(8000);
        turnOffAllSegments();
        digitalWrite(led2, LOW);
        break;
case 0:
        digit3();zero();
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        delay(8000);
        turnOffAllSegments();
        digitalWrite(led2, LOW);
        break;
default:
        digit3();
```

```
            turnOffAllSegments();
            digitalWrite(led2, LOW);
            break;
        }
  }
  num = -1;
  delay(1000);
}

void zero(){
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, HIGH);
digitalWrite(pinDP, HIGH);
  }

  void one(){
digitalWrite(pinA, HIGH);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, HIGH);
digitalWrite(pinDP, HIGH);
}

void two(){
  digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, LOW);
digitalWrite(pinDP, HIGH);
  }

  void three(){
        digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, LOW);
```

```
digitalWrite(pinDP, HIGH);
      }

 void four(){
      digitalWrite(pinA, HIGH);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
digitalWrite(pinDP, HIGH);
      }

 void five(){
      digitalWrite(pinA, LOW);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
digitalWrite(pinDP, HIGH);
      }

 void six(){
      digitalWrite(pinA, LOW);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
digitalWrite(pinDP, HIGH);
      }

 void seven(){
      digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, HIGH);
digitalWrite(pinDP, HIGH);
      }

 void eight(){
      digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
```

```
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
digitalWrite(pinDP, HIGH);
            }

  void nine(){
            digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
digitalWrite(pinDP, HIGH);
            }

  void digit3(){
            digitalWrite(D1, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
digitalWrite(D4, LOW);
            }

  void turnOffAllSegments(){
            digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, HIGH);
digitalWrite(pinDP, HIGH);
            }
```

3. SAMPLE INPUT AND OUTPUT

Consider the case when the transmitter sends out 5.
5 is read from the serial monitor and converted to binary by the transmitter.

0	1	0	1

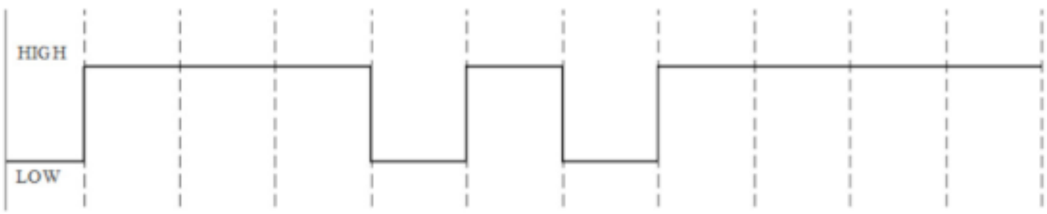
Fig: representation of 5 in the binary system.

To these data bits, 3 frame bits are added on each side. So the resulting frame is 10 bits and transmitted as:

1	1	1	0	1	0	1	1	1	1

The red bits represent the frame start and stop bits. This is transmitted one bit per second starting with the MSB.

Timing diagram:

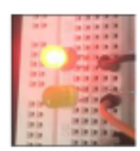


These pulses reach the receiver. The frame bits at the receiver's side are initialised to 0. As the pulses start arriving, the bits start shift to the left as shown:

At time t = 0s
No bit is received. All bits are in the initial state.

0	0	0	0	0	0	0	0	0	0

At time t =1s
The MSB of the start frame bit is received. All the remaining bits are shifted to the left by 1 and the leftmost bit is discarded. The red led lights up indicating the reception of a high signal.

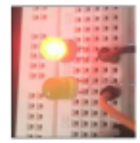


0	0	0	0	0	0	0	0	0	1

At time t =2s
The second bit of the start frame is received. The remaining bits are left-shifted as before and the red led lights up.

0	0	0	0	0	0	0	0	1	1

At time t = 3s
The third bit of the start frame is received. The remaining bits are left-shifted as above and red led lights up. At this point the start frame bits have arrived successfully.

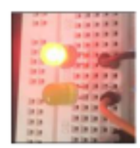


0	0	0	0	0	0	0	1	1	1

At time t = 4s
First data bit is received and the red led is turned off. The remaining bits are left shifted.

0	0	0	0	0	0	1	1	1	0

At time t = 5s
Second data bit is received. The remaining bits are left shifted. led is turned high.

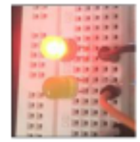


0	0	0	0	0	1	1	1	0	1

At time t = 6s
The third data bit is received. led is turned off. The remaining bits are left shifted.

0	0	0	0	1	1	1	0	1	0

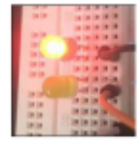
At time t = 7s



The third data bit is received. led is turned on. The remaining bits are left shifted. At this point all the data bits have been received successfully.

0	0	0	1	1	1	0	1	0	1

At time t = 8s



The first bit of the frame stop bits arrives. The red led lights up and the remaining bits are left-shifted.

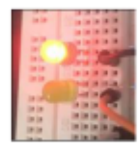
0	0	1	1	1	0	1	0	1	1

At time t = 9s

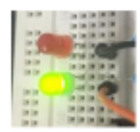
Same procedure as in t = 8s.

0	1	1	1	0	1	0	1	1	1

At time t = 10s



Same procedure as in t = 8s. However, at this point, the receiver detects that the first and last three bits are all high. This matched the frame sequence and thus has obtained a correct frame. The green led is set high.



The 4 middle bits are converted back to decimal and displayed on the 7-segment display. The receiver waits for 8 seconds and is again ready to receive a new frame.

1	1	1	0	1	0	1	1	1	1

Thus 5 is displayed on the 7-segment display

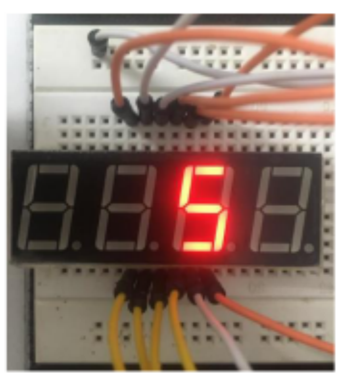


Fig: 4-dgiti 7-segment display with one active digit

A few more sample outputs:

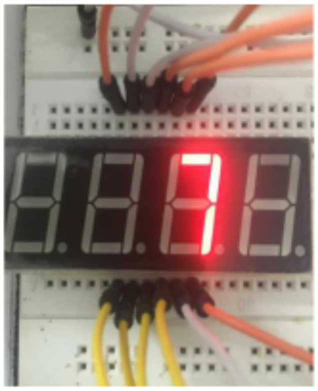


Fig : output = 0

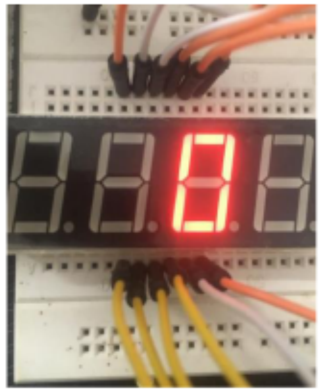


Fig : output = 7

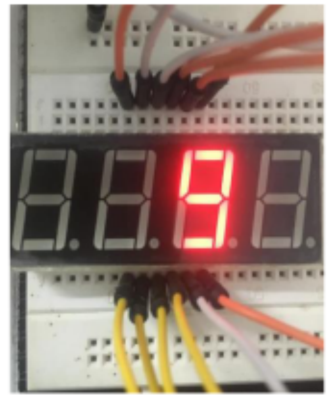


Fig : output = 9

4. APPLICATIONS

- All digital computing devices and electronics work on the principle of binary logic systems as they are easy to realise.
- Data of any form can be represented in binary form and can be converted back as an when required.
- Binary encoding forms the basis for all forms of data transfer including laser and optical fiber communication and transmission of data over networks.
- This point to point laser communication can be used in applications requiring the use of encryption or cryptography such as long-range communications between units in an army.

RESULTS

This project has demonstrated the use of Arduino boards, the implementation of a real time communication system, the interfacing of a 7-segment display and binary to decimal conversion and vice-versa using embedded C.