**Name : Nikita Kulkarni**

**Class : SE-IT**

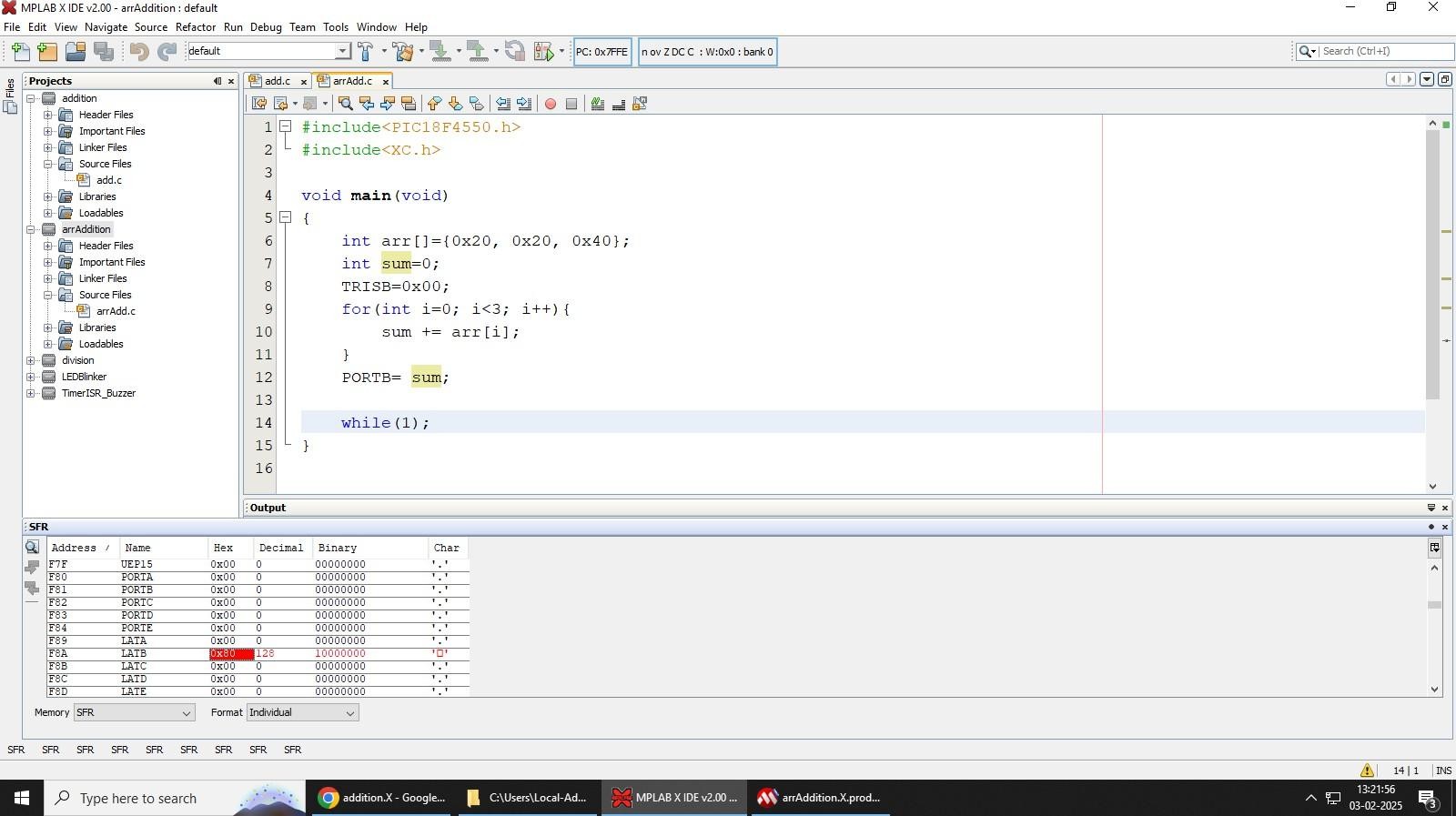
**Division : 1**

**Roll No. : 207A060**

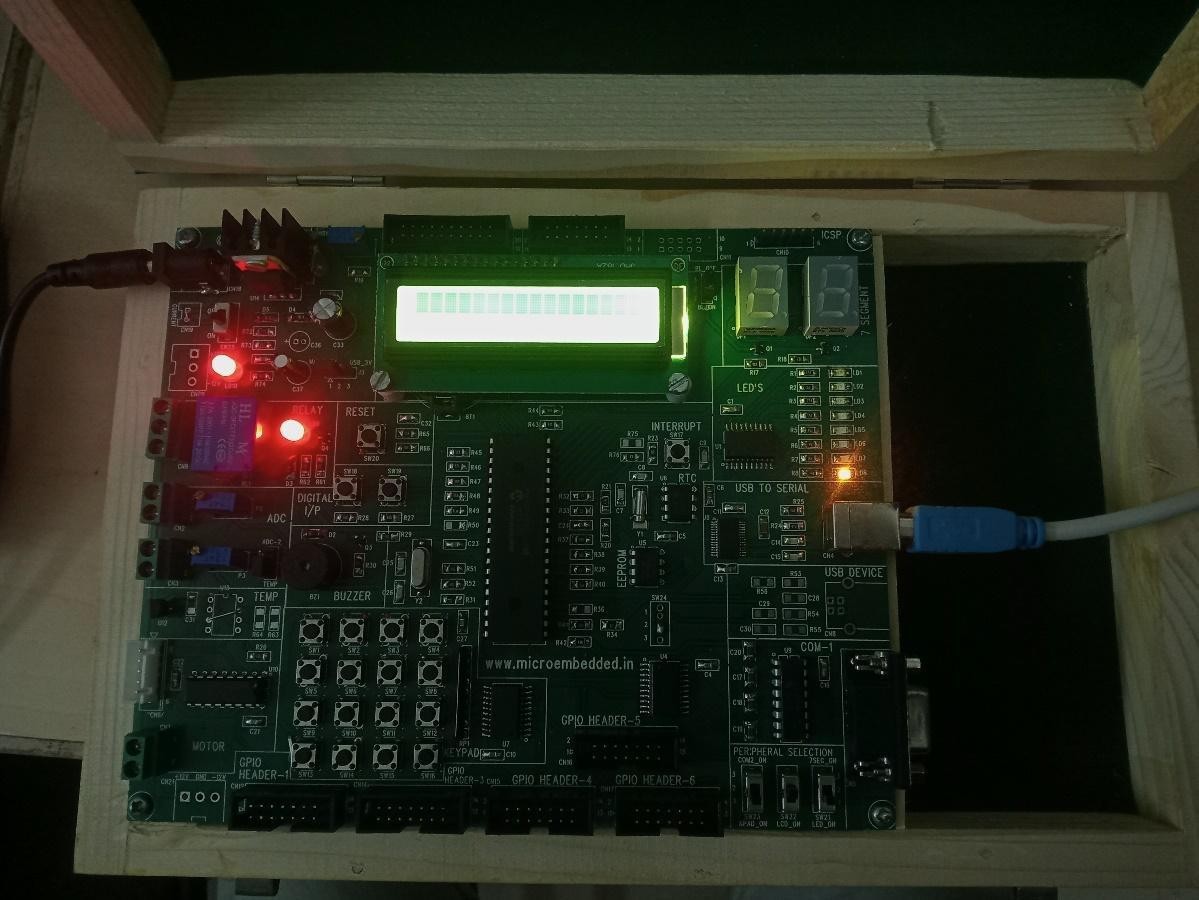
**Batch :**

**Experiment 01**: Write an Embedded C program to add array of n numbers.

**MAPLABX Interface (C Program and SFRs):**

****

**PIC18F4550 Kit Output:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

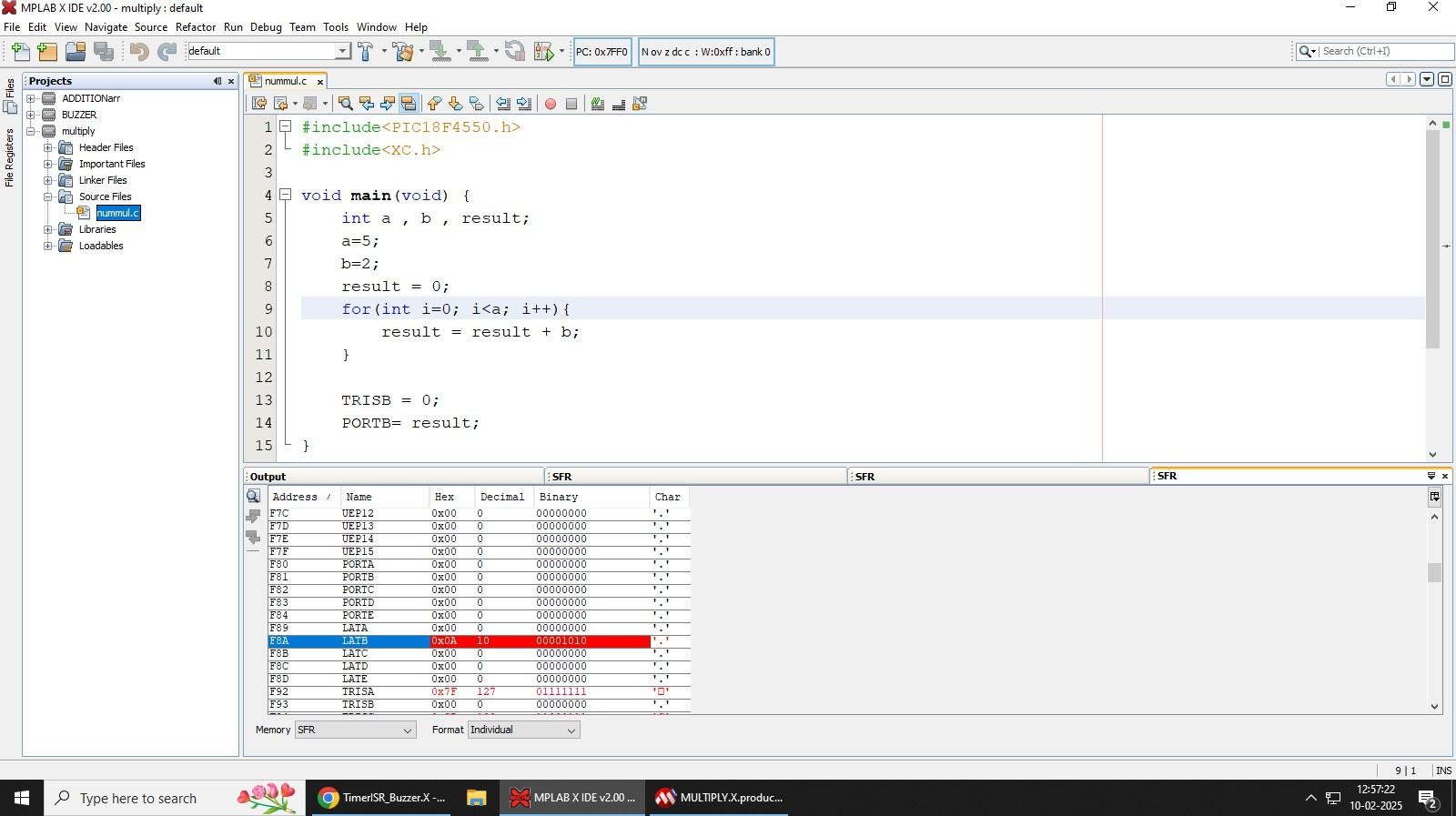
**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 02**: Write an Embedded C menu driven program for: i) Multiply 8-bit number by 8 bit number ii) Divide 8 bit number by 8 bit number

**MAPLABX Interface (C Program and SFRs) Multiplication:**

****

**PIC18F4550 Kit Output Multiplication:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

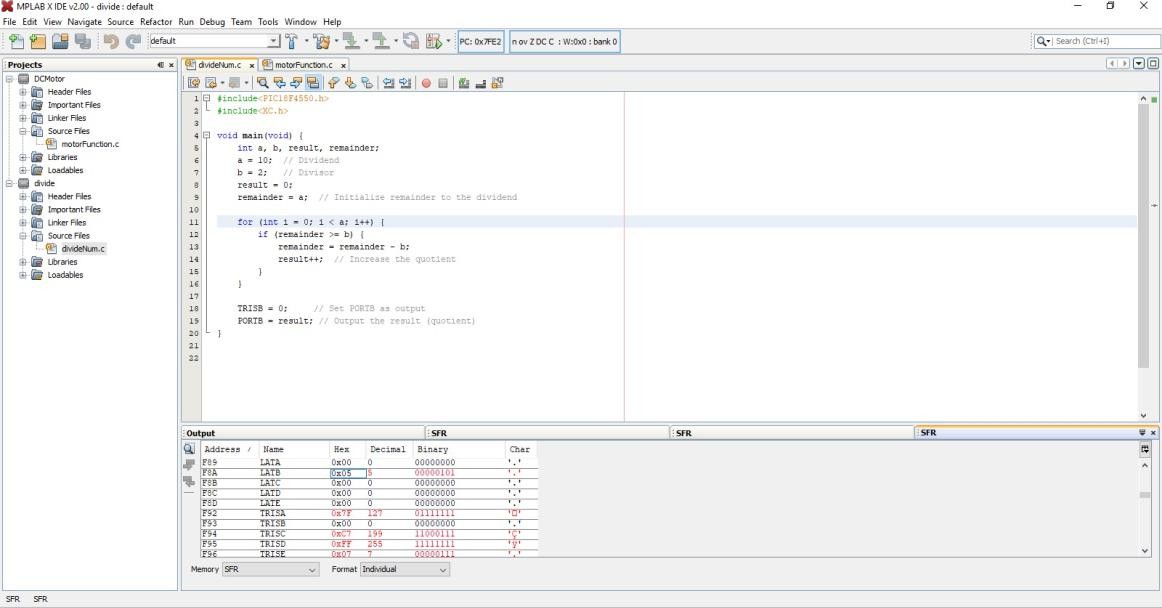
**Division : 1**

**Roll No. : 207A060**

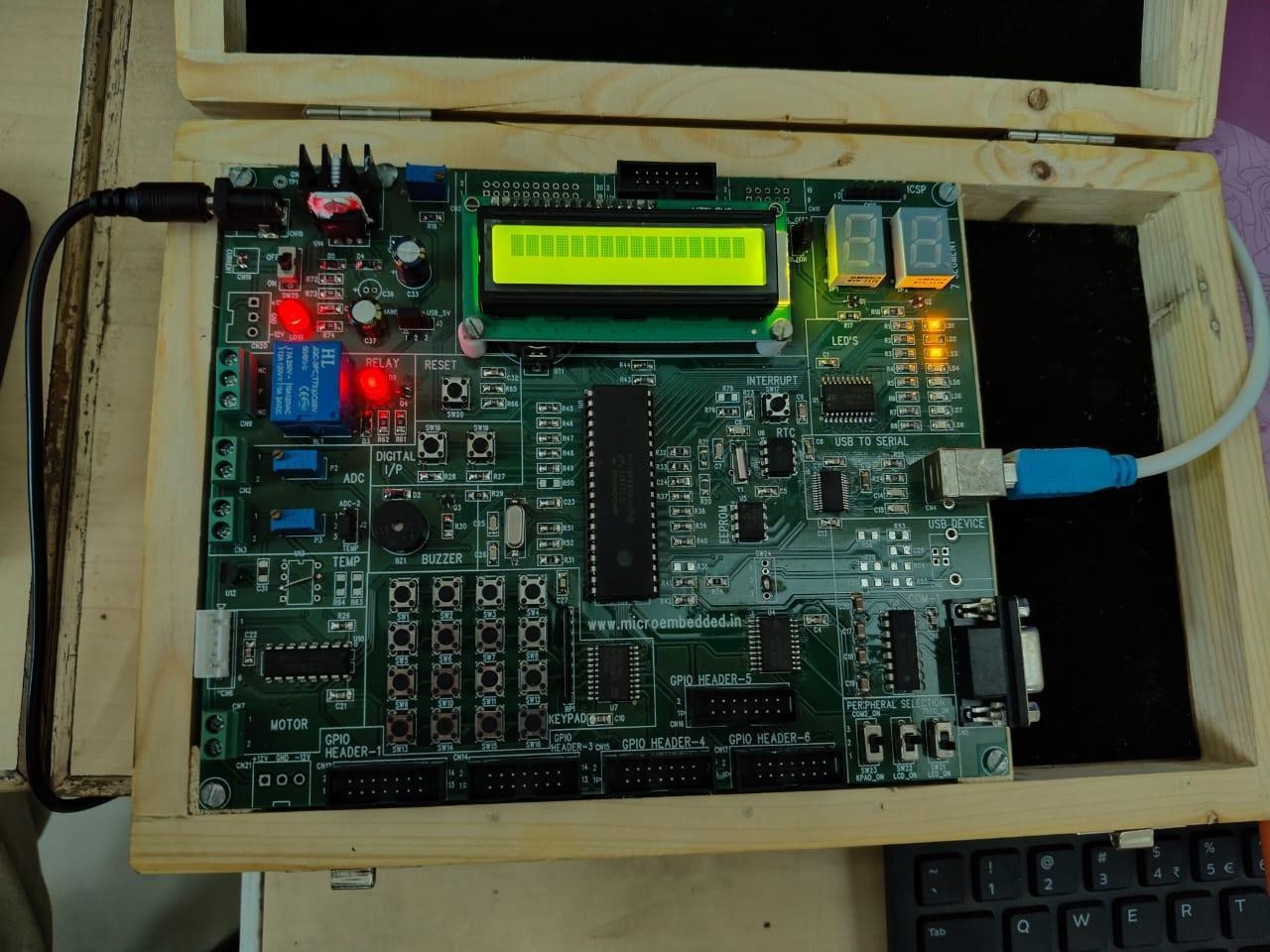
**Batch : C**

**Experiment 02**: Write an Embedded C menu driven program for: i) Multiply 8-bit number by 8 bit number ii) Divide 8 bit number by 8 bit number

**MAPLABX Interface (C Program and SFRs) Division:**

****

**PIC18F4550 Kit Output Division:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

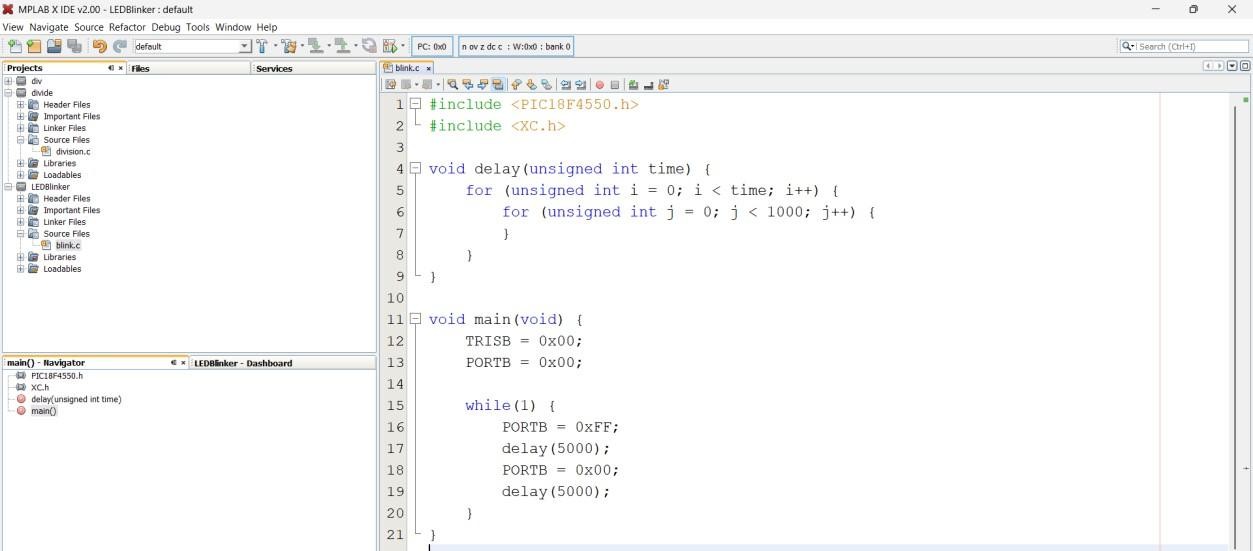
**Division : 1**

**Roll No. : 207A060**

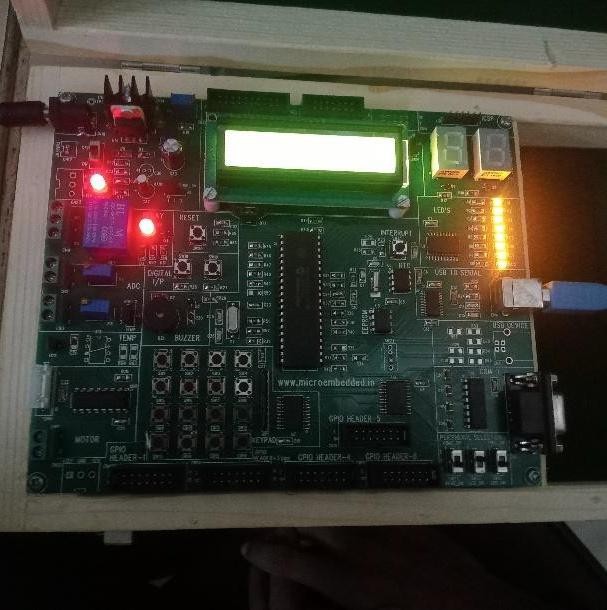
**Batch : C**

**Experiment 04**: Write an Embedded C program to interface PIC18F4550 with LED & blinking it using specified delay.

**MAPLABX Interface (C Program and SFRs):**

****

**PIC18F4550 Kit Output: LED Switches ON & OFF after delay:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 05S**: Write an Embedded C program for Timer programming ISR based buzzer on/off.

**C Program for buzzer sound generation:**

#include <pic18f4550.h> /\* Contains PIC18F4550 specifications \*/ #define Buzzer LATAbits.LATA5 /\* Define buzzer pin \*/ unsigned int count = 0;

void interrupt Timer1\_ISR()

{

if(TMR1IF==1)

{

//TMR1=0xCF2C; TMR1L = 0x20; TMR1H = 0xD1;

count ++;

if (count >= 1000) //measure upto 1000 ms i.e. 1 seconds

{

Buzzer = ~Buzzer; /\* Toggle buzzer pin \*/ count = 0; //reset count

}

TMR1IF = 0; //timer1 overflow flag to 0

}

}

void main()

{

TRISB=0; /\* Set as output port \*/ TRISAbits.TRISA5 = 0; //set buzzer pin RA5 as output GIE=1; /\* Enable Global Interrupt \*/

PEIE=1; /\* Enable Peripheral Interrupt \*/

TMR1IE=1; /\* Enable Timer1 Overflow Interrupt \*/

TMR1IF=0;

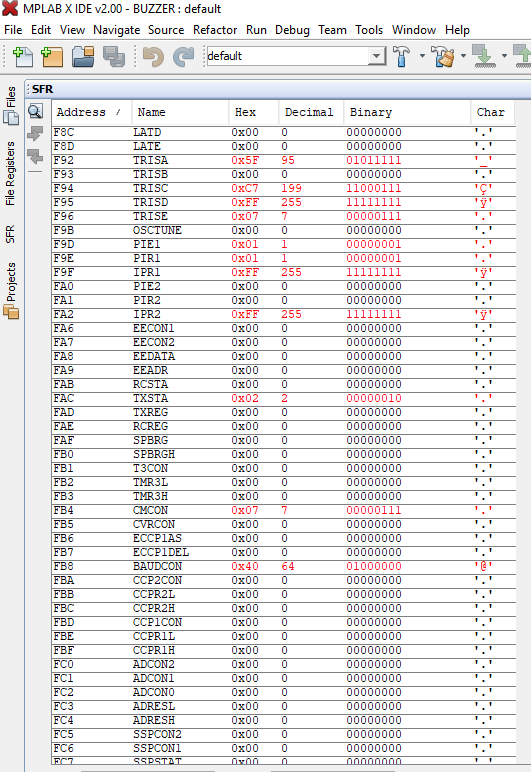
/\* Enable 16-bit TMR1 register,no pre-scale,internal clock, timer OFF \*/

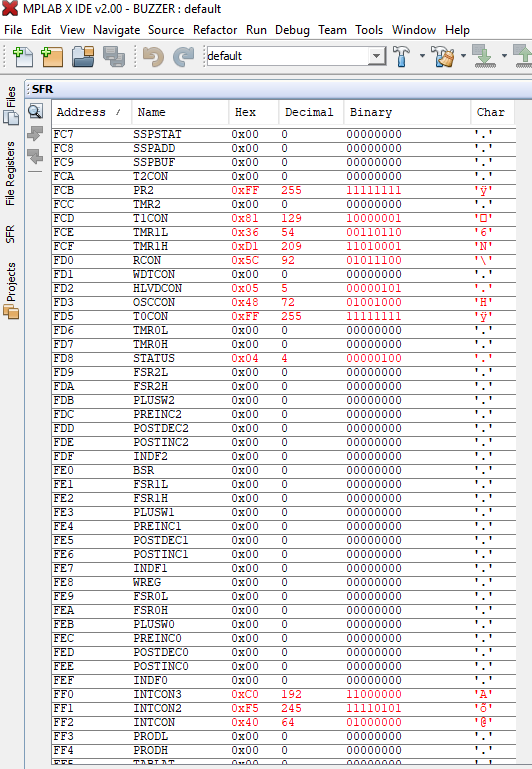
T1CON=0x80; /\*1:8 prescale\*/

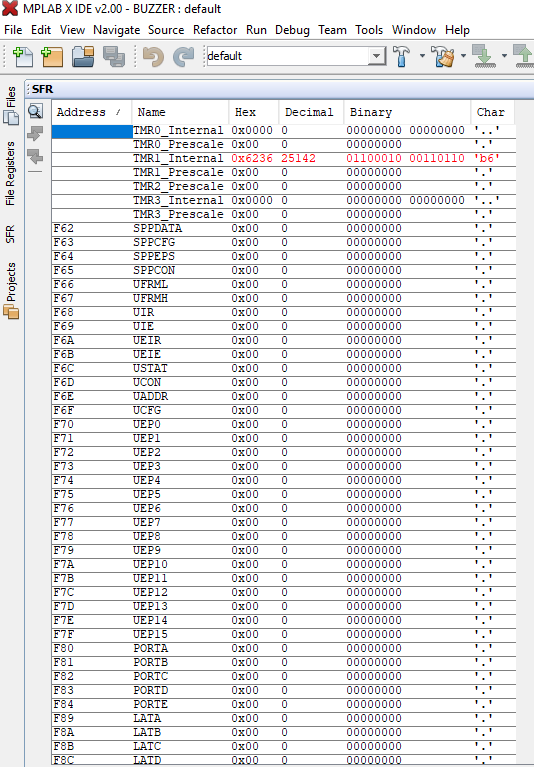
TMR1L = 0x20; TMR1H = 0xD1;

TMR1ON=1; /\* Turn ON Timer1 \*/

while(1);}

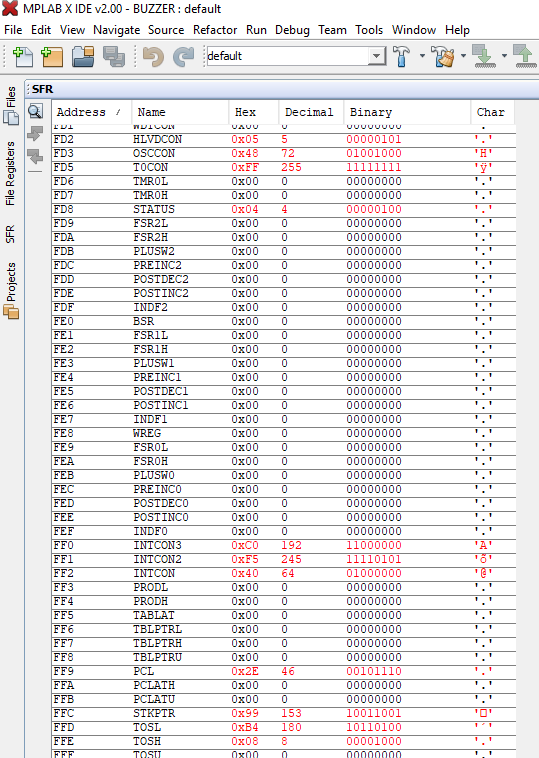
**PIC18F4550 Kit Output: Buzzer sound with delay of 1 second MAPLABX Interface (SFRs)**

****



**MAPLABX Interface (SFRs):**

`

****

**Name :Nikita Kulkarni**

**Class : SE-IT**

**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 06**: Write an Embedded C program for LCD interfacing with PIC 18FXXX.

**C Program for** LCD interfacing with PIC 18F4550**:**

#include <p18f4550.h>

#define LCD\_EN LATAbits.LA1 #define LCD\_RS LATAbits.LA0 #define LCDPORT LATB

void lcd\_delay(unsigned int time)

{

unsigned int i , j ;

for(i = 0; i < time; i++)

{

for(j=0;j<100;j++);

}

}

void SendInstruction(unsigned char command)

{

LCD\_RS = 0; // RS low : Instruction LCDPORT = command;

LCD\_EN = 1; // EN High lcd\_delay(10);

LCD\_EN = 0; // EN Low; command sampled at EN falling edge

lcd\_delay(10);

LCDPORT = lcddata;

LCD\_EN = 1; // EN High lcd\_delay(10);

LCD\_EN = 0; // EN Low; data sampled at EN falling edge

lcd\_delay(10);

}

void InitLCD(void)

{

ADCON1 = 0x0F;

TRISB = 0x00; //set data port as output TRISAbits.RA0 = 0; //RS pin TRISAbits.RA1 = 0; // EN pin

SendInstruction(0x38); //8 bit mode, 2 line,5x7 dots SendInstruction(0x06); // entry mode SendInstruction(0x0C); //Display ON cursor OFF SendInstruction(0x01); //Clear display SendInstruction(0x80); //set address to 1st line

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*/

unsigned char \*String1 = "Microembedded"; unsigned char \*String2 = "PIC-18F Board";

}

void SendData(unsigned char lcddata)

{

LCD\_RS = 1; // RS HIGH : DATA

void main(void)

{

ADCON1 = 0x0F;

TRISB = 0x00; //set data port as output

TRISAbits.RA0 = 0; //RS pin TRISAbits.RA1 = 0; // EN pin

SendInstruction(0x38); //8 bit mode, 2 line,5x7 dots SendInstruction(0x06); // entry mode SendInstruction(0x0C); //Display ON cursor OFF SendInstruction(0x01); //Clear display SendInstruction(0x80); //set address to 1st line

while(\*String1)

{

SendData(\*String1); String1++;

}

SendInstruction(0xC0); //set address to 2nd line while(\*String2)

{

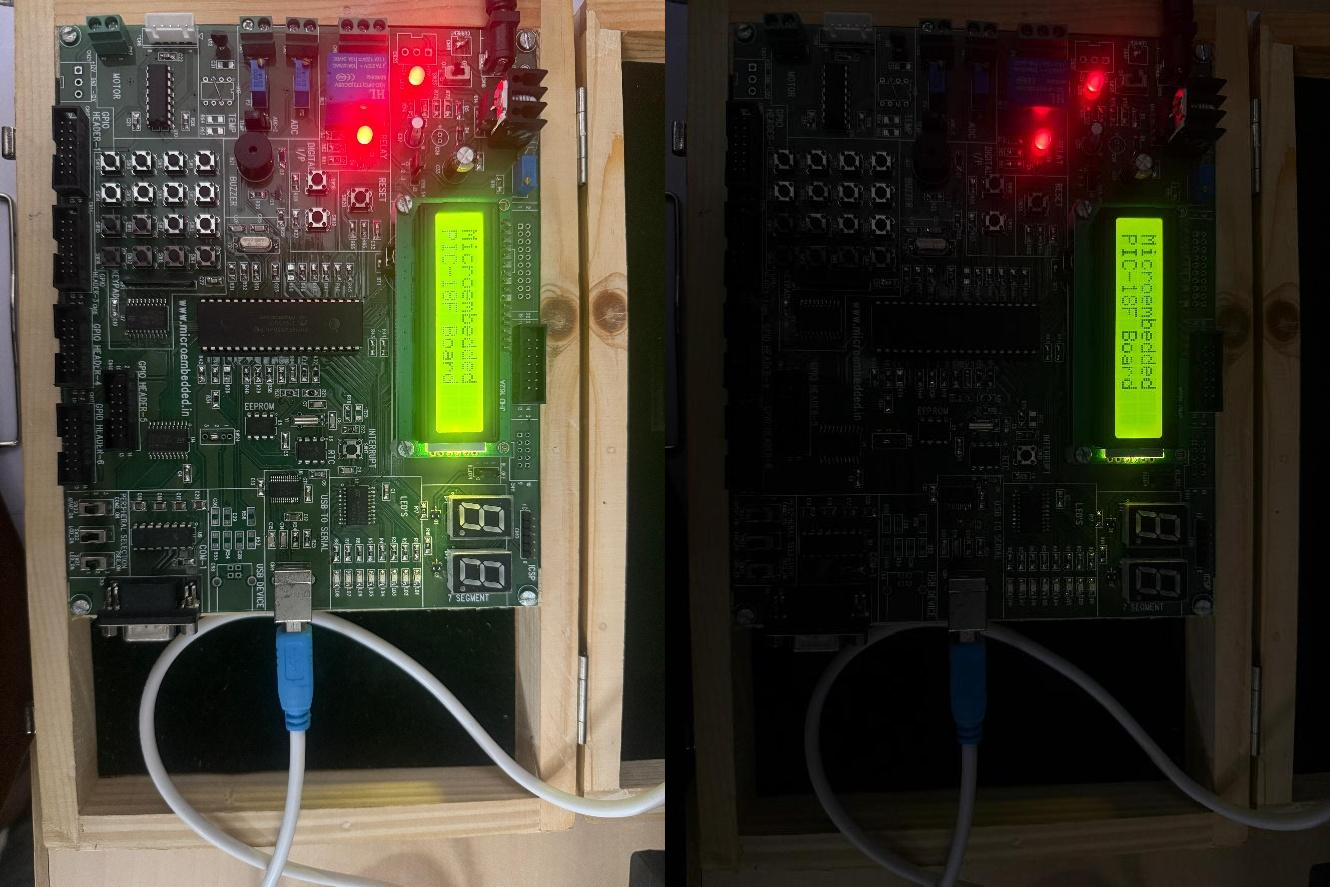
SendData(\*String2); String2++;

}

while(1);

}

**PIC18F4550 Kit Output:**

****

**Name : Nikita kulkarni**

**Class : SE-IT**

**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 07**: Write an Embedded C program for Generating PWM signal for servo motor/DC motor.

**C program for Generating PWM signal for servo motor/DC motor:**

/\* Calculations

* Fosc = 48MHz

\*

* PWM Period = [(PR2) + 1] \* 4 \* TMR2 Prescale Value / Fosc
* PWM Period = 200us
* TMR2 Prescale = 16
* Hence, PR2 = 149 or 0x95
* Duty Cycle = 10% of 200us
* Duty Cycle = 20us
* Duty Cycle = (CCPR1L:CCP1CON<5:4>) \* TMR2 Prescale Value / Fosc
* CCP1CON<5:4> = <1:1>
* Hence, CCPR1L = 15 or 0x0F

\*/

#include<p18f4550.h>

unsigned char count=0; bit TIMER,SPEED\_UP;

void timer2Init(void)

{

T2CON = 0b00000010; //Prescalar = 16; Timer2 OFF PR2 = 0x95; //Period Register

}

void delay(unsigned int time)

{

unsigned int i,j; for(i=0;i<time;i++)

for(j=0;j<1000;j++);

}

void main(void)

{

|  |  |  |
| --- | --- | --- |
| unsigned int i; |  |  |
| TRISCbits.TRISC1 | = 0; | //RC1 pin as output |
| TRISCbits.TRISC2 | = 0; | //CCP1 pin as output |
| LATCbits.LATC1 | = 0; |  |

CCP1CON = 0b00111100; //Select PWM mode; Duty cycle LSB CCP1CON<4:5> = <1:1> CCPR1L =

0x0F; //Duty cycle 10%

timer2Init(); //Initialise Timer2

TMR2ON = 1; //Timer2 ON

while(1) //Loop forever

{

for(i=15;i<150;i++)

{

CCPR1L = i;

delay(100);

}

for(i=150;i>15;i--)

{

CCPR1L = i;

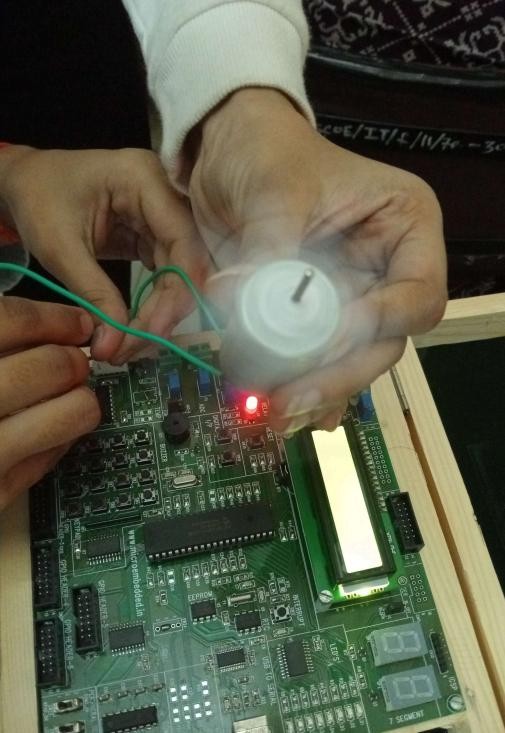
delay(100);

}

}

}

**PIC18F4550 Kit Output:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 08**: Write an Embedded C program for PC to PC serial communication using UART.

**C program for PC to PC serial communication using UART**

/\*Baud Rate GENERATION

* n => required baudrate
* BRGH = 0
* SPBRG = (Fosc / (64 \* n)) -1
* For 9600 baudrate, SPBRG ~=77

\*/

#include<p18F4550.h> #include<stdio.h>

#define Fosc 48000000UL

void InitUART(unsigned int baudrate)

{

TXREG = data; //Transmit data

}

void putch(unsigned char data)

{

SendChar(data);

}

unsigned char GetChar(void)

{

while(!PIR1bits.RCIF); //Wait till receive buffer becomes full

return RCREG; //Returned received data

TRISCbits.RC6 = 0; //TX pin set as output

TRISCbits.RC7 = 1; //RX pin set as input

SPBRG = (unsigned char)(((Fosc /64)/baudrate)-1);

BAUDCON = 0b00000000; //Non-inverted data; 8- bit baudrate generator

TXSTA = 0b00100000; //Asynchronous 8-bit; Transmit enabled; Low speed baudrate select

RCSTA = 0b10010000; //Serial port enabled; 8-bit data; single receive enabled

}

void SendChar(unsigned char data)

{

while(TXSTAbits.TRMT == 0); //Wait while transmit register is empty

}

void main(void)

{

InitUART(9600);

printf("\r\nHello MicroPIC-18F: Enter any Key from Keyboard\r\n");

while(1)

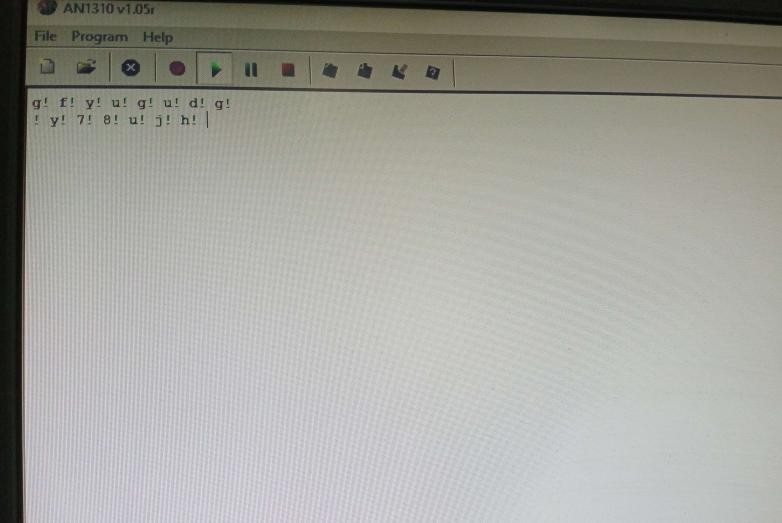
{

printf("%c! ",GetChar()); //Receive character from PC and echo back

}

while(1); }

**Output:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 09**: Write an Embedded C program for Temperature sensor interfacing using ADC & display on LCD.

**C program for for Temperature sensor interfacing using ADC & display on LCD.**

/\* }

* File: main.c }
* Author: MicroEmbedded
* ​
* Created on October 15, 2016, 4:35 PM

\*/

#include <pic18f4550.h> #include <stdio.h>

#define LCD\_EN LATAbits.LA1 #define LCD\_RS LATAbits.LA0 #define LCDPORT LATB

void SendInstruction(unsigned char command)

{

LCD\_RS = 0;

low : Instruction LCDPORT = command; LCD\_EN = 1;

High

lcd\_delay(10); LCD\_EN = 0;

// RS

// EN

// EN

unsigned char str[16];

void lcd\_delay(unsigned int time)

{

unsigned int i , j ;

for(i = 0; i < time; i++)

{

for(j=0;j<100;j++);

Low; command sampled at EN falling edge lcd\_delay(10);

}

void SendData(unsigned char lcddata)

{

LCD\_RS = 1;

// RS

HIGH : DATA

LCDPORT = lcddata;

LCD\_EN = 1;

High

lcd\_delay(10); LCD\_EN = 0;

Low; data sampled at EN falling edge

// EN

// EN

while(\*ch) SendData(\*ch++);

}

lcd\_delay(10);

}

void InitLCD(void)

{

ADCON1 = 0x0F;

TRISB = 0x00; //set data port as output TRISAbits.RA0 = 0; //RS pin

TRISAbits.RA1 = 0; // EN pin

SendInstruction(0x38); //8 bit mode, 2 line,5x7 dots

SendInstruction(0x06);

//entr

y mode

SendInstruction(0x0C);

void ADCInit(void)

{

TRISEbits.RE2 = 1; //ADC channel 7 input

ADCON1 = 0b00000111; //Ref

voltages Vdd & Vss; AN0 - AN7 channels Analog

ADCON2 = 0b10101110; //Right

justified; Acquisition time 4T; Conversion clock Fosc/64

}

unsigned short Read\_Temp(void)

{

ADCON0 = 0b00011101; //ADC on; Select channel;

lay ON cursor OFF

//Disp

GODONE = 1; //Start Conversion

SendInstruction(0x01); //Clear display SendInstruction(0x80); //set address to 0

}

void LCD\_display(unsigned int row, unsigned int pos, unsigned char \*ch)

{

if(row==1)

SendInstruction(0x80 | (pos-1)); else

SendInstruction(0xC0 | (pos-1));

while(GO\_DONE == 1 ); //Wait till A/D conversion is complete

return ADRES; //Return ADC result

}

int main(void)

{

unsigned int temp; InitLCD(); ADCInit();

LCD\_display(1,1,"Temperature:"); while(1)

{

temp = Read\_Temp();

temp = ((temp \* 500) / 1023); sprintf(str,"%d'C ",temp);

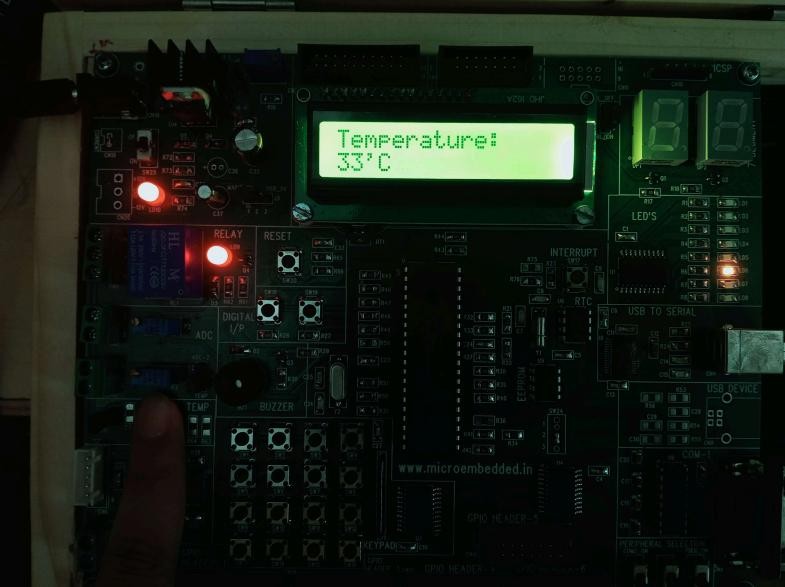
LCD\_display(2,1,str); lcd\_delay(9000);

}

return 0;

}

**Output:**

****

**Name : Nikita Kulkarni**

**Class : SE-IT**

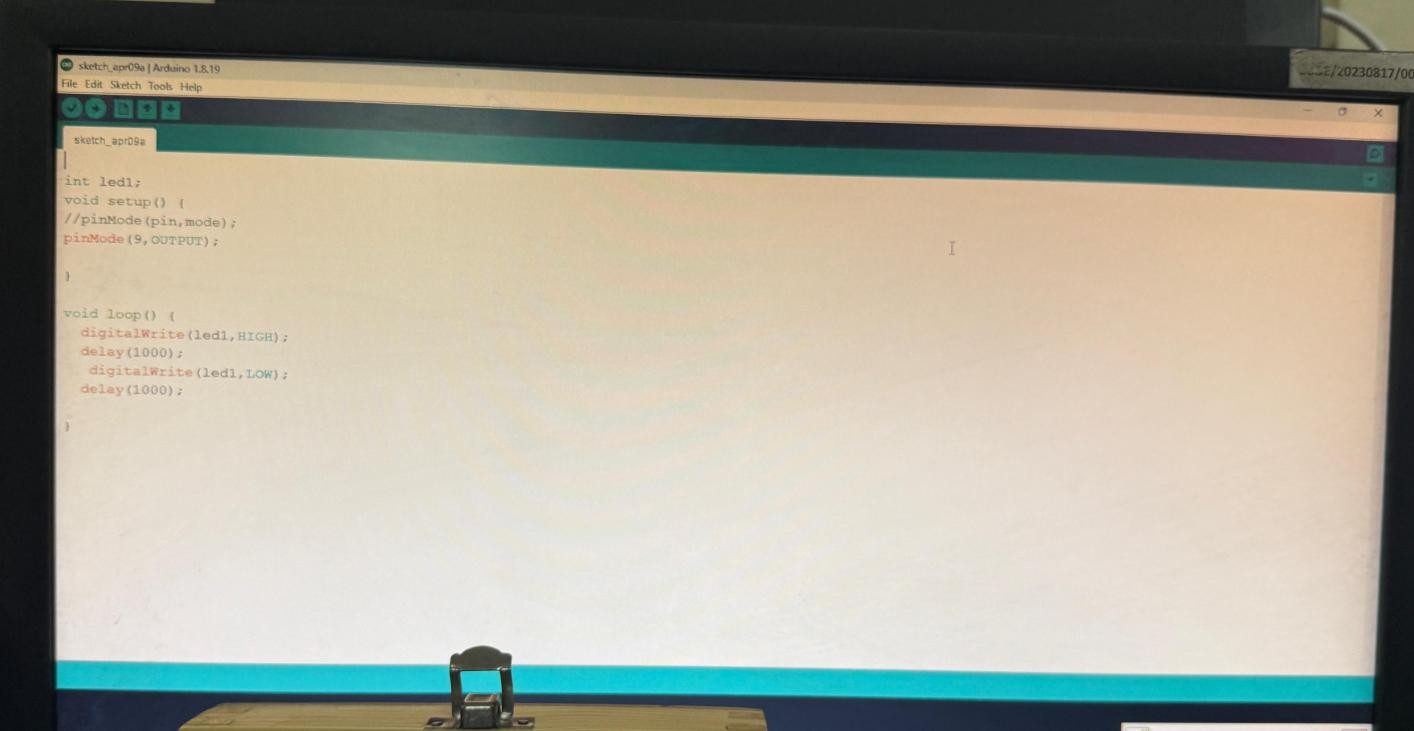
**Division : 1**

**Roll No. : 207A060**

**Batch : C**

**Experiment 10:** Write simple program using Open-source prototype platform like Arduino for digital read/write using LED and switch, Analog read/write using sensor and actuators.

**Output:**

****

