**// Experiment 5: Interfacing LEDs, Switches, Buzzer, and Relay**

**// Name: Dnyaneshwar Arjun Kharachane**

**// Class: TE(A) Roll No: 3401054**

#include <p18f4520.h> // Include Controller specific header

#include <delays.h>

#define BUZZER PORTAbits.RA3 // Buzzer connected to PORTA 3rd PIN

#define SWITCH0 PORTBbits.RB0 // Switch0 connected to PORTB 0th PIN

#define SWITCH1 PORTBbits.RB1 // Switch1 connected to PORTB 1st PIN

void main(void) {

TRISA = 0x00; // RA3, Output Direction

TRISB = 0xFF; // RB0, B1 Input Direction

TRISD = 0x00; // [RD0-3=LEDs] [RD4,5=Relay1,2] Output Direction

PORTD = 0xFF; // [RD0-3=LEDs] [RD4,5=Relay1,2] Initialize as 0xFF

while (1) {

if (!SWITCH1) { // Condition for 1st switch

while (1) {

BUZZER = 1; // Buzzer On

PORTD = 0x37; // (Relay1=1, Relay2=1) & (LEDs sequence Left to Right=0111=7)

Delay10KTCYx(100); // 400ms Delay

PORTD = 0x3B; // (LEDs sequence Left to Right=1011=B)

Delay10KTCYx(100);

PORTD = 0x3D;

Delay10KTCYx(100);

PORTD = 0x3E;

Delay10KTCYx(100);

if (!SWITCH0) // Check if 2nd switch is pressed

break;

}

} else if (!SWITCH0) { // Condition for 2nd switch

while (1) {

BUZZER = 0; // Buzzer Off

PORTD = 0xCE; // (Relay1=0, Relay2=0) & (LEDs sequence Right to Left=1110=E)

Delay10KTCYx(100);

PORTD = 0xCD; // LEDs sequence Right to Left=1101=D

Delay10KTCYx(100);

PORTD = 0xCB;

Delay10KTCYx(100);

PORTD = 0xC7;

Delay10KTCYx(100);

if (!SWITCH1) // Check if 1st switch is pressed

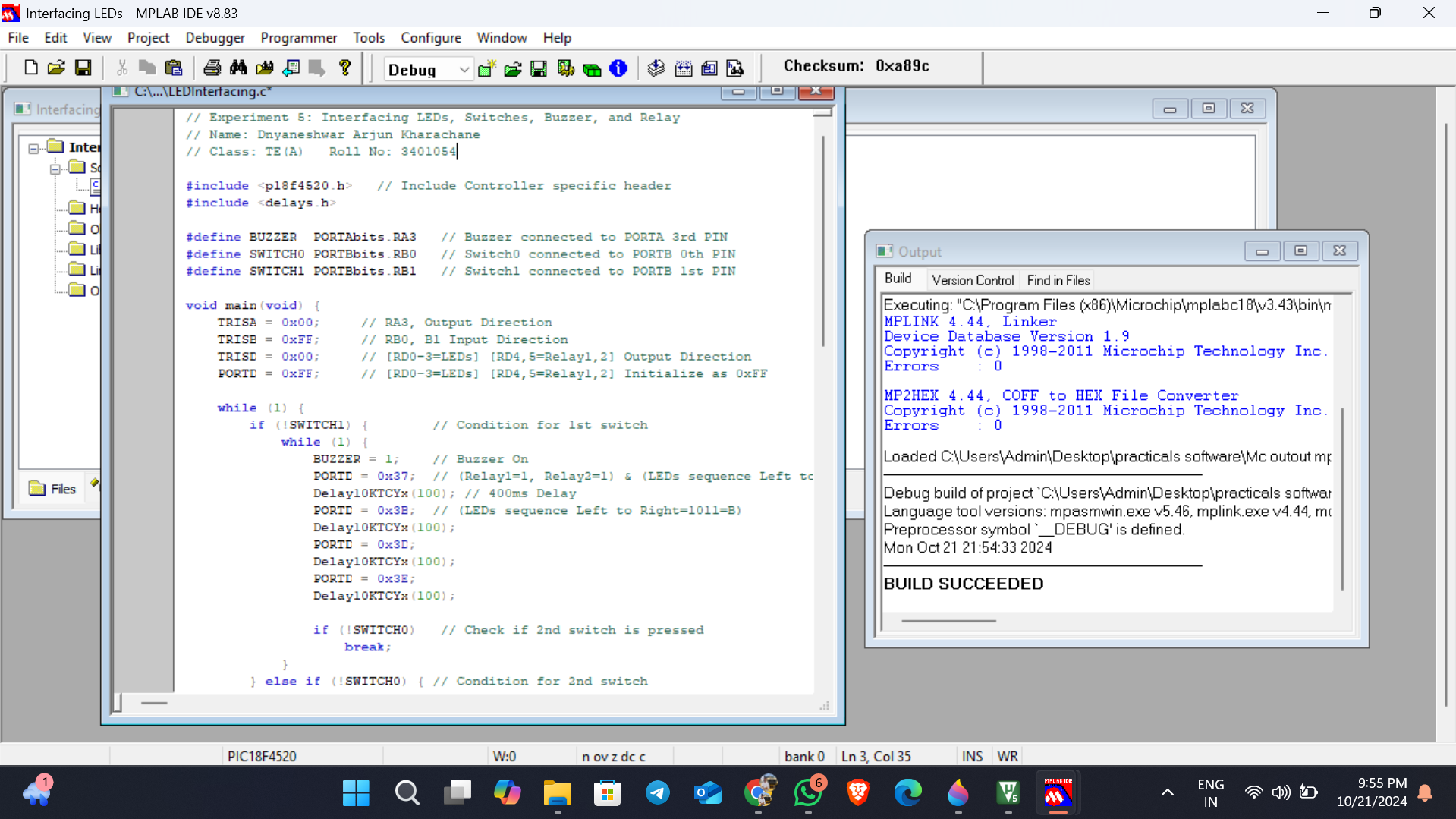
break;

}

}

}

}



**// Experiment 6: Interfacing 16x2 LCD with PIC18F Microcontroller**

**// Name: Dnyaneshwar Arjun Kharachane**

**// Class: TE(A) Roll No: 3401054**

#include <p18f4520.h>

// Configuration Bits

#pragma config OSC = HS

#pragma config PWRT = OFF

#pragma config WDT = OFF

#pragma config DEBUG = OFF, LVP = OFF

// Function Prototype declaration

void lcdcmd(unsigned char value);

void lcddata(unsigned char value);

void msdelay(unsigned int itime);

// Define LCD Control Pins

#define ldata PORTD // Declare ldata variable for PORTD

#define rs PORTEbits.RE0 // Declare rs variable for pin RE0

#define rw PORTEbits.RE1 // Declare rw variable for pin RE1

#define en PORTEbits.RE2 // Declare en variable for pin RE2

void main() {

TRISD = 0x00; // Set direction of PORTD as output

TRISE = 0x00; // Set direction of PORTE as output

msdelay(50);

lcdcmd(0x38); // 16x2 LCD

msdelay(50);

lcdcmd(0x0E); // Display on, Cursor on

msdelay(15);

lcdcmd(0x01); // Clear Display screen

msdelay(15);

lcdcmd(0x06); // Increment cursor and shift right

msdelay(15);

lcdcmd(0x80); // Force cursor on the first row, first position

lcddata('S'); // Display character 'S'

msdelay(50);

lcddata('P'); // Display character 'P'

msdelay(50);

lcddata('P'); // Display character 'P'

msdelay(50);

lcddata('U'); // Display character 'U'

msdelay(50);

lcdcmd(0xC0); // Move cursor to the next line

msdelay(15);

lcddata('S'); // Display character 'S'

msdelay(50);

lcddata('I'); // Display character 'I'

msdelay(50);

lcddata('T'); // Display character 'T'

msdelay(50);

lcddata('S'); // Display character 'S'

msdelay(50);

}

void lcdcmd(unsigned char value) {

ldata = value; // Send the command value to PORTD

rs = 0; // Command register is selected

rw = 0; // Write mode is selected

en = 1; // Generate High to Low pulse on Enable pin

msdelay(1);

en = 0;

}

void lcddata(unsigned char value) {

ldata = value; // Send the data value to PORTD

rs = 1; // Data register is selected

rw = 0; // Write mode is selected

en = 1; // Generate High to Low pulse on Enable pin

msdelay(1);

en = 0;

}

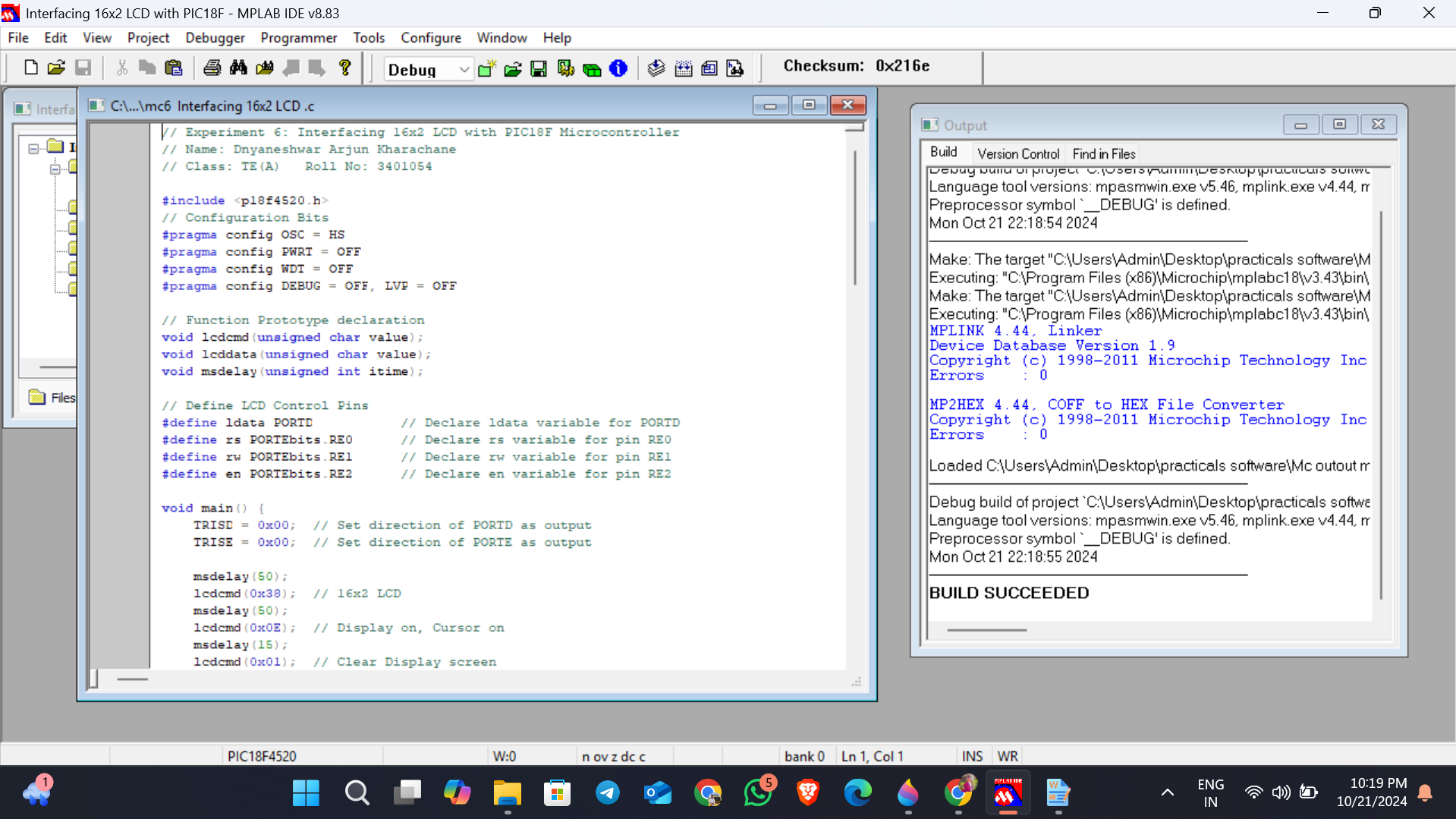
void msdelay(unsigned int itime) {

int i, j;

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

for (j = 0; j < 135; j++);

}



**// Experiment 7: To generate square wave using timer of PIC18F4550 with interrupt.**

**// Name: Dnyaneshwar Arjun Kharachane**

**// Class: TE(A) Roll No: 3401054**

#include <P18F4550.h>

void delay\_timer() {

T0CON = 0x08; // Timer 0, 16-bit mode, no prescaler

TMR0H = 0xFB; // Values calculated for 0.1 ms delay

TMR0L = 0x50;

T0CONbits.TMR0ON = 1; // Timer 0 On

while (INTCONbits.TMR0IF == 0); // Wait until TMR0IF gets flagged

T0CONbits.TMR0ON = 0; // Timer 0 Off

INTCONbits.TMR0IF = 0; // Clear Timer 0 interrupt flag

}

void main() {

TRISCbits.TRISC0 = 0; // Configure RC0 as output

while (1) {

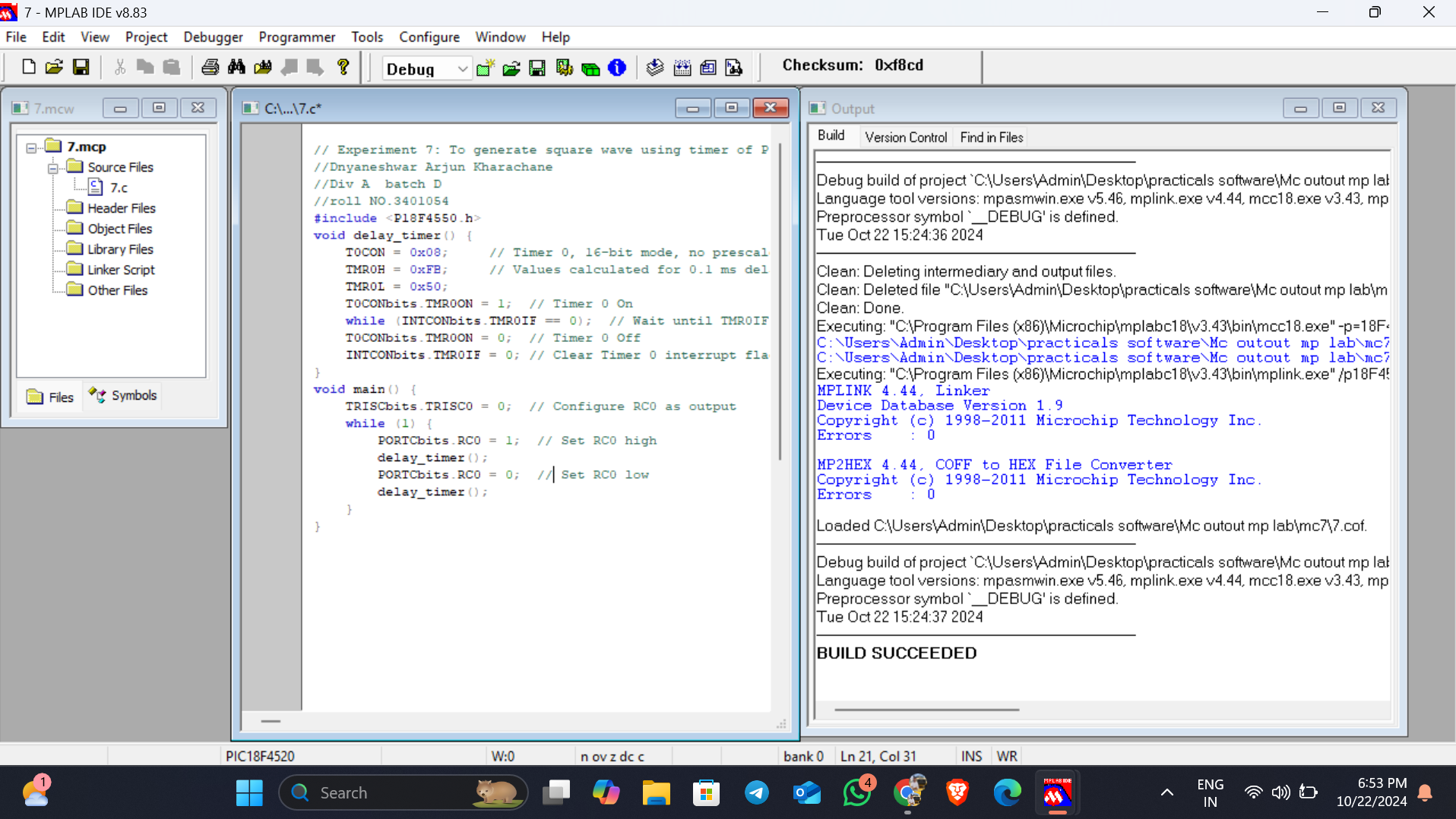
PORTCbits.RC0 = 1; // Set RC0 high

delay\_timer();

PORTCbits.RC0 = 0; // Set RC0 low

delay\_timer();

}  
}



**// Experiment 8: Interfacing serial port with PC both side communication.**

**// Name: Dnyaneshwar Arjun Kharachane**

**// Class: TE(A) Roll No: 3401054**

#include <P18F4520.h>

void main(void) {

TXSTA = 0x20; // Choose low baud rate, 8-bit mode

SPBRG = 15; // 9600 baud rate with XTAL 10 MHz

TXSTAbits.TXEN = 1; // Enable transmission

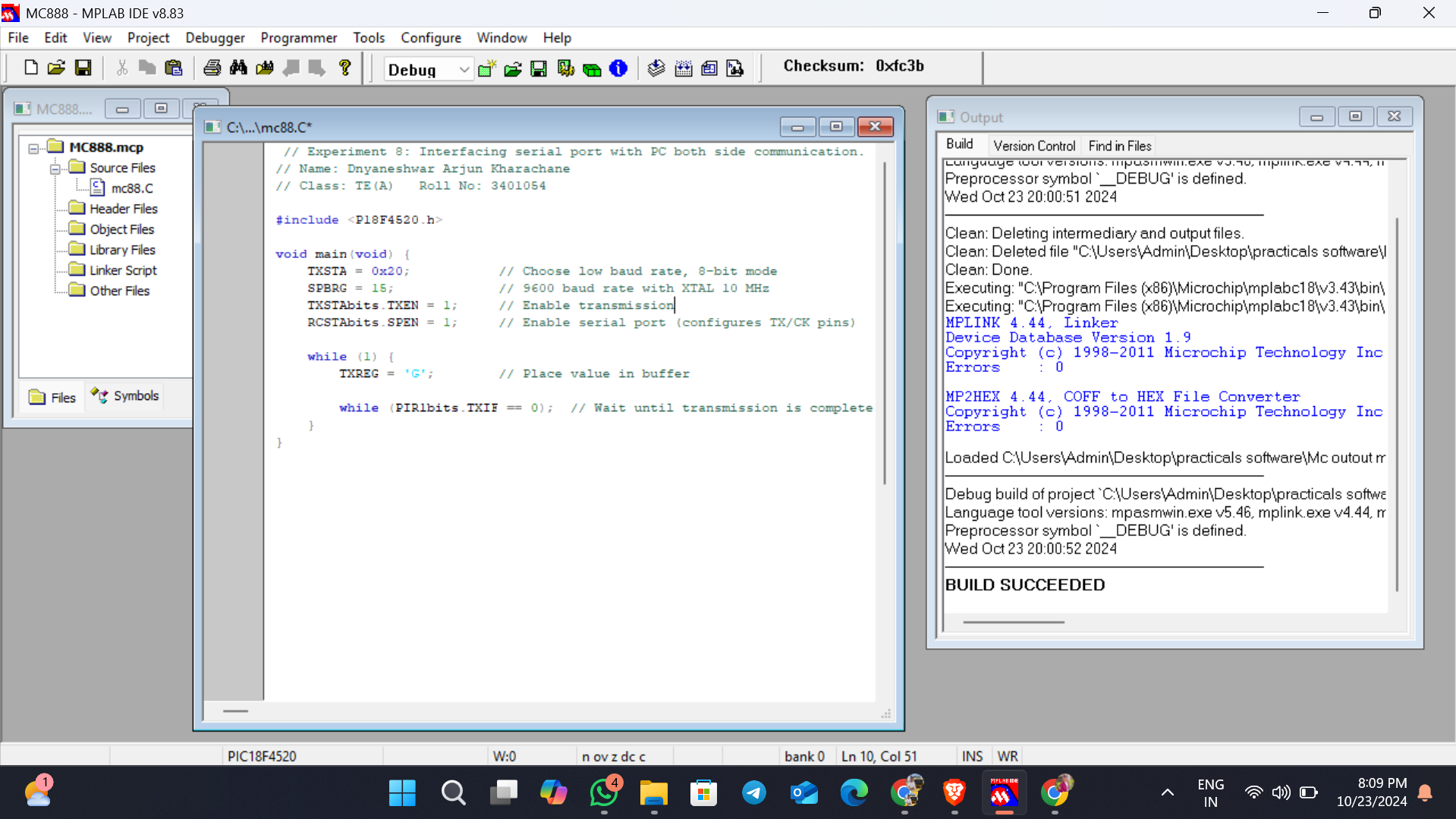
RCSTAbits.SPEN = 1; // Enable serial port (configures TX/CK pins)

while (1) {

TXREG = 'G'; // Place value in buffer

while (PIR1bits.TXIF == 0); // Wait until transmission is complete

}

}  


**// Experiment 9: Generation of PWM Signal for DC Motor Control.**

**// Name: Dnyaneshwar Arjun Kharachane**

**// Class: TE(A) Roll No: 3401054**

#include <p18f458.h>

#define SW PORTDbits.RD7 // Define SW as the pin RD7

#define ENABLE PORTBbits.RB0 // Define ENABLE as pin RB0

#define MTR\_1 PORTBbits.RB1 // Define MTR\_1 as pin RB1

#define MTR\_2 PORTBbits.RB2 // Define MTR\_2 as pin RB2

void main() {

TRISD = 0x80; // Make RD7 an input pin

TRISB = 0x00; // Make PORTB output

SW = 1;

ENABLE = 0;

MTR\_1 = 0;

MTR\_2 = 0;

while (1) {

ENABLE = 1;

if (SW == 1) {

MTR\_1 = 1;

MTR\_2 = 0;

} else {

MTR\_1 = 0;

MTR\_2 = 1;

}

}

}  
