

```

#include <LiquidCrystal.h> // Library for 16x2 LCD Display
const int rs = 8, en = 7, d4 = 6, d5 = 5, d6 = 4, d7 = 3; // initialize the library by associating any needed LCD interface pin with the IC pin number it is connected to
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Function declarations
int DTMF(); // Function to read a number key pressed using a DTMF Decoder
int PressAnyKey(); // Function to know if any key is pressed
char CorrectPasscode(); // Function to read next input (* or #) if user passcode matched to preset one
char Lock(); // Function to read # if user pressed it to lock the device

// Global variable declarations
int DTMFdata; // Initialize the DTMF data variable for decimal of 4 bit binary.
int DTMFvalue; // Variable to indicate the value of the actual key pressed for DTMF()
char DTMFvalue1; // Variable to indicate the value of the actual key pressed for CorrectPasscode()
char DTMFvalue2; // Variable to indicate the value of the actual key pressed for Lock()
int StQ = 9; // DTMF Module StQ Pin connects to IC Digital Pin 9
int Q4 = 10; // DTMF Module Q4 Pin connects to IC Digital Pin 10
int Q3 = 11; // DTMF Module Q3 Pin connects to IC Digital Pin 11
int Q2 = 12; // DTMF Module Q2 Pin connects to IC Digital Pin 12
int Q1 = 13; // DTMF Module Q1 Pin connects to IC Digital Pin 13
int Passcode[4] = {0, 0, 0, 0}; // Array for storing the passcode

// void setup() starts
void setup()
{
    lcd.begin(16, 2);
    pinMode(Q1, INPUT); // Declaring the pinModes of other pins of the IC that is used
    pinMode(Q2, INPUT); // Declaring the pinModes of other pins of the IC that is used
    pinMode(Q3, INPUT); // Declaring the pinModes of other pins of the IC that is used
    pinMode(Q4, INPUT); // Declaring the pinModes of other pins of the IC that is used
    pinMode(StQ, INPUT); // Declaring the pinModes of other pins of the IC that is used
    pinMode(2, OUTPUT); // Declaring the pinModes of other pins of the IC that is used
    pinMode(1, OUTPUT); // Declaring the pinModes of other pins of the IC that is used
}
// void setup() ends

// void loop() starts
void loop()
{
    // Variable declarations
    int Entered_Passcode[4]; // Array to store the passcode input by user
    int Any_Key = 0; // Variable used to acknowledge if any key was pressed
    int Compare = 0; // Variable for comparison of user entered passcode
    int i,j,k,l; // Variables for 'for loop'
    char NextInput; // Variable to collect next input
    int Lock1 = 2; // Variable storing digital pin number to which lock is connected
    int LED = 1; // Variable storing digital pin number to which LED Green is connected
    char LockInput; // Variable to collect # to lock device
    // Variable declarations ends

    // Idle Mode starts
    // Code to print the Idle mode display "Press any key_"
    lcd.home();
    lcd.print("Press any key");
    // Code to print the Idle mode display "Press any key_" ends
    // Device leaving Idle mode....

    // If statement for unlocking or resetting the passcode starts
    Any_Key = PressAnyKey();
    delay(100);
    if(Any_Key == 1)
    {
        // LCD code to print "Enter the passcode:" on display
        lcd.clear();
        lcd.home();
        lcd.print("Enter the pass:");
        lcd.setCursor(0,1);
        lcd.cursor();
        // LCD code to print "Enter the passcode:" on display ends

        // for loop to obtain passcode from user to Entered_Passcode variable
        for(i=0 ; i<4 ; i++)
        {
            Entered_Passcode[i] = DTMF();
            // Code to print "*" on display when a number is pressed
            lcd.setCursor(i,1);
            lcd.print('*');
            delay(100);
            // Code to print "*" on display when a number is pressed ends
        }
        // for loop to obtain passcode from user to Entered_Passcode variable ends

        // for loop to compare passcode from user to preset passcode
        for(j=0 ; j<4 ; j++)
        {
            if(Passcode[j] == Entered_Passcode[j])
            {
                Compare++;
            }
        }
        // for loop to compare passcode from user to preset passcode ends

        // if statement to do further proceedings if the entered passcode is correct
        if(Compare == 4)
        {
            // Code to print instructions on LCD and wait for next input
            lcd.clear();
            lcd.home();
            lcd.noCursor();
            lcd.print("Correct Passcode");
            delay(1000);
            lcd.clear();
            lcd.home();
            lcd.print("Press # - UNLOCK");
            lcd.setCursor(1,1);
            lcd.print("* - CHANGE PASS");
            NextInput = CorrectPasscode();
            delay(500);
            // Code to print instructions on LCD and wait for next input ends

            switch(NextInput)

```

```

{
    case '*':
        // LCD code to enter new passcode
        lcd.clear();
        lcd.home();
        lcd.print("Enter new pass:");
        lcd.setCursor(0,1);
        lcd.cursor();
        for(k=0 ; k<4 ; k++)
        {
            Passcode[k] = DTMF();
            // Code to print "*" on display when a number is pressed
            lcd.setCursor(k,1);
            lcd.print('*');
            delay(100);
            // Code to print "*" on display when a number is pressed ends
        }
        lcd.clear();
        lcd.home();
        lcd.noCursor();
        lcd.print("New passcode set");
        for(l=0 ; l<4 ; l++)
        {
            lcd.setCursor(l,1);
            lcd.print(Passcode[l]);
        }
        delay(1000);
        // LCD code to enter new passcode ends
        lcd.clear();
        goto End;
        break;
    case '#':
        // LCD code to unlock device
        digitalWrite(Lock1, HIGH);
        digitalWrite(LED, HIGH);
        lcd.clear();
        lcd.home();
        lcd.print("UNLOCKED!");
        delay(1000);
        // LCD code to unlock device ends
        // LCD code to lock device
        lcd.clear();
        lcd.home();
        lcd.print("Press # to LOCK");
        LockInput = Lock();
        delay(500);
        if(LockInput == '#')
        {
            lcd.clear();
            lcd.home();
            lcd.print("LOCKED!");
            digitalWrite(Lock1, LOW);
            digitalWrite(LED, LOW);
            delay(1000);
            // LCD code to lock device ends
            lcd.clear();
            goto End;
        }
        break;
    }
}
// if statement to do further proceedings if the entered passcode is correct ends

// else statement to do further proceedings if the entered passcode is incorrect
else
{
    lcd.clear();
    lcd.home();
    lcd.noCursor();
    lcd.print("Wrong Passcode!");
    delay(1000);
    lcd.clear();
    goto End;
}
// else statement to do further proceedings if the entered passcode is incorrect ends
}
// If statement for unlocking or resetting the passcode ends
End:
delay(0);
}
// void loop() ends

// DTMF Function Starts
int DTMF()
{
    // IF StQ is High a DTMF tone is present. Check 4-bit code.
    // While loop starts
    while(1)
    {
        DTMFdata = 0;
        DTMFvalue = '\0';
        // If statement starts
        if (digitalRead(StQ) == HIGH)
        {
            if (digitalRead(Q1) == HIGH) // If Q1 High add 1 to DTMFdata
            {
                DTMFdata = DTMFdata + 1;
            }
            if (digitalRead(Q2) == HIGH) // If Q2 High add 2 to DTMFdata
            {
                DTMFdata = DTMFdata + 2;
            }
            if (digitalRead(Q3) == HIGH) // If Q3 High add 4 to DTMFdata
            {
                DTMFdata = DTMFdata + 4;
            }
            if (digitalRead(Q4) == HIGH) // If Q4 High add 8 to DTMFdata
            {
                DTMFdata = DTMFdata + 8;
            }
        }
    }
    // decode 4-bit code. Which button was pressed?

```

```

// Switch statement starts
switch (DTMFdata)
{
    case 0: //D
        continue;
        break;
    case 1: //1
        DTMFvalue = 1;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 2: //2
        DTMFvalue = 2;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 3: //3
        DTMFvalue = 3;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 4: //4
        DTMFvalue = 4;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 5: //5
        DTMFvalue = 5;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 6: //6
        DTMFvalue = 6;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 7: //7
        DTMFvalue = 7;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 8: //8
        DTMFvalue = 8;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 9: //9
        DTMFvalue = 9;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 10: //0
        DTMFvalue = 0;        // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 11: // *
        continue;
        break;
    case 12: // #
        continue;
        break;
    case 13: // A
        continue;
        break;
    case 14: // B
        continue;
        break;
    case 15: // C
        continue;
        break;
}
// Switch statement ends
break;        // To break the infinite while loop
}
// If statement ends
}
// While loop ends
return DTMFvalue;    // Return the value stored in variable DTMFvalue
}
// DTMF Function ends

// PressAnyKey Function starts
int PressAnyKey()
{
    // While loop starts
    while(1)
    {
        // If statement starts
        if (digitalRead(StQ) == HIGH)
        {
            break;        // To break the infinite while loop
        }
        // If statement ends
    }
    // While loop ends
    return (1);    // Returns 1
}
// PressAnyKey Function ends

// CorrectPasscode Function Starts
char CorrectPasscode()
{
    // IF StQ is High a DTMF tone is present. Check 4-bit code.
    // While loop starts
    while(1)
    {
        DTMFdata = 0;
        DTMFvalue1 = '\0';
        // If statement starts
        if (digitalRead(StQ) == HIGH)
        {
            if (digitalRead(Q1) == HIGH) // If Q1 High add 1 to DTMFdata
            {
                DTMFdata = DTMFdata + 1;
            }
            if (digitalRead(Q2) == HIGH) // If Q2 High add 2 to DTMFdata
            {
                DTMFdata = DTMFdata + 2;
            }
            if (digitalRead(Q3) == HIGH) // If Q3 High add 4 to DTMFdata
            {
                DTMFdata = DTMFdata + 4;
            }
            if (digitalRead(Q4) == HIGH) // If Q4 High add 8 to DTMFdata
            {
                DTMFdata = DTMFdata + 8;
            }
        }

        //decode 4-bit code. Which button was pressed?
        // Switch statement starts
        switch (DTMFdata)

```

```

    {
        case 0: //D
            continue;
            break;
        case 1: //1
            continue;
            break;
        case 2: //2
            continue;
            break;
        case 3: //3
            continue;
            break;
        case 4: //4
            continue;
            break;
        case 5: //5
            continue;
            break;
        case 6: //6
            continue;
            break;
        case 7: //7
            continue;
            break;
        case 8: //8
            continue;
            break;
        case 9: //9
            continue;
            break;
        case 10: //0
            continue;
            break;
        case 11: // *
            DTMFvalue1 = '*'; // Setting the value of the actual key pressed to DTMFvalue variable
            break;
        case 12: // #
            DTMFvalue1 = '#'; // Setting the value of the actual key pressed to DTMFvalue variable
            break;
        case 13: //A
            continue;
            break;
        case 14: //B
            continue;
            break;
        case 15: //C
            continue;
            break;
    }
    // Switch statement ends
    break; // To break the infinite while loop
}
// If statement ends
}
// While loop ends
return DTMFvalue1; // Return the value stored in variable DTMFvalue
}
// CorrectPasscode Function ends

// Lock Function Starts
char Lock()
{
    // IF StQ is High a DTMF tone is present. Check 4-bit code.
    // While loop starts
    while(1)
    {
        DTMFdata = 0;
        DTMFvalue2 = '\0';
        // If statement starts
        if (digitalRead(StQ) == HIGH)
        {
            if (digitalRead(Q1) == HIGH) // If Q1 High add 1 to DTMFdata
            {
                DTMFdata = DTMFdata + 1;
            }
            if (digitalRead(Q2) == HIGH) // If Q2 High add 2 to DTMFdata
            {
                DTMFdata = DTMFdata + 2;
            }
            if (digitalRead(Q3) == HIGH) // If Q3 High add 4 to DTMFdata
            {
                DTMFdata = DTMFdata + 4;
            }
            if (digitalRead(Q4) == HIGH) // If Q4 High add 8 to DTMFdata
            {
                DTMFdata = DTMFdata + 8;
            }
        }

        //decode 4-bit code. Which button was pressed?
        // Switch statement starts
        switch (DTMFdata)
        {
            case 0: //D
                continue;
                break;
            case 1: //1
                continue;
                break;
            case 2: //2
                continue;
                break;
            case 3: //3
                continue;
                break;
            case 4: //4
                continue;
                break;
            case 5: //5
                continue;
                break;
            case 6: //6

```

```

        continue;
        break;
    case 7: //7
        continue;
        break;
    case 8: //8
        continue;
        break;
    case 9: //9
        continue;
        break;
    case 10: //0
        continue;
        break;
    case 11: // *
        continue;
        break;
    case 12: // #
        DTMFvalue2 = '#'; // Setting the value of the actual key pressed to DTMFvalue variable
        break;
    case 13: //A
        continue;
        break;
    case 14: //B
        continue;
        break;
    case 15: //C
        continue;
        break;
    }
    // Switch statement ends
    break; // To break the infinite while loop
}
// If statement ends
}
// While loop ends
return DTMFvalue2; // Return the value stored in variable DTMFvalue
}
// Lock Function ends

```