|  |
| --- |
| #include <LiquidCrystal.h>  // Library Code for LCD Display  LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Initialize the library with the arduino pins  float tempsensor; // The variable we will use to store the sensor input  float temp; // The variable we will use to store temperature in degrees.  float HCUltra=0; // To input position values  const int buzzertemp = 10; // Buzzer to arduino pin 10    void setup()  {  pinMode(6, OUTPUT); // 6th pin to green led  pinMode(7, OUTPUT); // 7th pin to red led  lcd.begin(16, 2); // Set up LCD's collumns and rows    }  void loop()  {    tempsensor = analogRead(A0); // Read the analog sensor and store it  temp = tempsensor / 1024; // Find percentage of input reading  temp = temp \* 5; // Multiply by 5V to get voltage  temp = temp - 0.5; // Subtract the offset  temp = temp \* 100; // Convert to degrees  HCUltra = calcDistance(9,8); // Call HC-SR04 Distance Calculator Function    if (HCUltra > 20 ) // Condition for distance of user from Sensor  {  blinking(7,6); // Call function to blink LEDs  updateDisplay(); // Call function to refresh LCD Display  lcd.setCursor(0,0);  lcd.print("Hello Friend!"); // Display message on first line  lcd.setCursor(0,1);  lcd.print("Come closer"); // Display message on second line  noTone(buzzertemp); // Buzzer makes no sound  }  else  {  if(temp < 37.5) // Condition for safe temperature  {  digitalWrite(6, HIGH); // Energize the green led  digitalWrite(7,LOW); // De-engergize the red led    updateDisplay();  lcd.setCursor(0, 0);  lcd.print("Temp: ");  lcd.print(temp);  lcd.print("C"); // Display the user's temperature in Celsius    lcd.setCursor(0,1);  lcd.print("You may enter!"); // Display welcome message  noTone(buzzertemp);  }  else  {  dangerBlink(7); // Call Function to blink red LED  digitalWrite(6,LOW); // De-engergize the green LED    updateDisplay();  lcd.setCursor(0, 0);  lcd.print("Temp: ");  lcd.print(temp);  lcd.print("C"); // Display the user's temperature in Celsius    lcd.setCursor(0,1);  lcd.print("DANGER! DANGER!"); // Display unsafe message  tone(buzzertemp,500); // Buzzer makes sound at 500Hz  }  }    }  float calcDistance(int trig1, int echo1) // Function to calculate distance  {  pinMode(trig1, OUTPUT);  pinMode(echo1, INPUT);  digitalWrite(trig1, LOW); // Set trigger pin low  delayMicroseconds(2);  digitalWrite(trig1, HIGH); // Set trigger pin high  delayMicroseconds(10);  digitalWrite(trig1, LOW); // Ping has now been sent  return pulseIn(echo1,HIGH,30000)/58.0; // Formula to get final distance in cm  }  void blinking(int pin6,int pin7) // Blinking LED Function  {  digitalWrite(pin7, HIGH); // Energise Red LED  delay(1000); // Wait for 1000 millisecond(s)  digitalWrite(pin7, LOW); // de-energise Red LED    digitalWrite(pin6, HIGH); // Energise Green LED  delay(1000); // Wait for 1000 millisecond(s)  digitalWrite(pin6, LOW); // De-energise Green LED    }  void updateDisplay() // LCD Display Refresh Function  {  delay(500);  lcd.clear(); // Refresh LCD Display every 0.5 seconds  }  void dangerBlink(int pin7)  {  digitalWrite(pin7, HIGH); // Energise Red LED  delay(200); // Wait for 200 millisecond(s)  digitalWrite(pin7, LOW); // de-energise Red LED  } |