```
/*****************************************
*
*   ECE342_Junior_Design_Temperature_Project
*
*   Uses devices to sense if a user is present,
*   measure their temperature, determine if they
*   have a fever and alert the user if they have one,
*   log user measurement information and time, and
*   shut down if no user input after five minutes.
*
*   Written by Cole Ferguson
*
*****************************************/
#include "LiquidCrystal_I2C.h"
#include<SD.h>
#include<SPI.h>
#include<Wire.h>
#include "SparkFun_VCNL4040_Arduino_Library.h"
#include <SparkFunMLX90614.h>
#include "LowPower.h"
#include <TimeLib.h>

IRTherm thermal;
VCNL4040 proximity;
File myFile;
LiquidCrystal_I2C lcd(0x27,18,2);
tmElements_t tm;

#define wakeupPin 2
#define buzzer 8
#define numRows 2
#define numCols 16
#define Temp_Fever 100.4
#define Samples 30

double User_Temp;
double Temp_Array[Samples];
void fevertone();
void healthytone();
int lastWakeup;

const char *monthName[12] = {
  "Jan", "Feb", "Mar", "Apr", "May", "Jun",
  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"
};

/*****************************************
* Utility function for printing preceding colon and leading zeros to SD card
*****************************************/
void fileprintDigits(int digits){
  myFile.print(":");
  if(digits < 10)
    myFile.print('0');
  myFile.print(digits);
}

/*****************************************
* Function for converting time AVR Macro strings converted to integers
*****************************************/
bool getTime(const char *str){
  int Hour, Min, Sec;
  if (sscanf(str, "%d:%d:%d", &Hour, &Min, &Sec) != 3) return false;
```

```
  tm.Hour = Hour;
  tm.Minute = Min;
  tm.Second = Sec;
  return true;
}

/*****************************************
* Function for saving date AVR Macro strings
*****************************************/
bool getDate(const char *str){
  char Month[12];
  int Day, Year;
  uint8_t monthIndex;

  if (sscanf(str, "%s %d %d", Month, &Day, &Year) != 3) return false;
  for (monthIndex = 0; monthIndex < 12; monthIndex++) {
    if (strcmp(Month, monthName[monthIndex]) == 0) break;
  }
  if (monthIndex >= 12) return false;
  tm.Day = Day;
  tm.Month = monthIndex + 1;
  tm.Year = CalendarYrToTm(Year);
  return true;
}

/*****************************************
* Function for setting new previous user interaction time
*****************************************/
void wakeup(){
  lastWakeup = minute();
  lcd.backlight();
  Serial.println(F("Waking up"));
  LCDlines("Waking up       ", "                ");
  thermal.wake();
  delay(1000);
}

/*****************************************
* Function for setting up devices, pin modes, and protocols used in the system
*****************************************/
void setup() {

  Serial.begin(115200);
  Wire.begin(); //Initialize I2C protocol bus

//Set Arduino system time to current time
  if (getDate(__DATE__) && getTime(__TIME__))
    Serial.println(F("AVR Macro strings converted to tmElements."));
  setTime(makeTime(tm));//set Ardino system clock to compiled time


//Initialize devices
  lcd.init(); //Initialize LCD screen
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0);

  if (thermal.begin() == false){ // Initialize thermal IR sensor
    Serial.println(F("Qwiic IR thermometer did not acknowledge! Freezing!"));
     while(1);
  }
  Serial.println(F("Qwiic IR Thermometer connected..."));
```

```
  thermal.setUnit(TEMP_F);
      delay(500);

  if (proximity.begin() == false){ // Initialize proximity sensor
    Serial.println(F("Device not found. Please check wiring."));
    while (1);
  }

  if (SD.begin(4) == false){ //Initialize SD card adapter
    Serial.println(F("SD card initialization failed"));
    while(1);
  }

//Set pin assignments and default values
  pinMode(buzzer,OUTPUT);
  lastWakeup = minute();
  //
  pinMode(wakeupPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(wakeupPin), wakeup, FALLING);
}

/*****************************************
* Function for printing strings to the LCD screen
*****************************************/
void LCDlines(String line1, String line2){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(line1);
    lcd.setCursor(0,1);
    lcd.print(line2);
    delay(300);
}

/*****************************************
* Function for measuring user temperature using the thermal sensor
*****************************************/
void getTemp(){
  for(int i = 0; i < Samples; i++){
      if(thermal.read()){//reads the user temperature several times and stores them in an
array
        Temp_Array[i] = thermal.object();
         }
      }
      for(int j = 0; j < Samples-1; j++){
        User_Temp += Temp_Array[j];
      }
      User_Temp = User_Temp / Samples;                //averages out values to find the user's
temperature
      User_Temp += 9;                                 //temperature offset/calibration
}

/*****************************************
* Function for testing whether the user has a fever/temperature above 100.4*F and playing
appropriate tone
*****************************************/
void testFever(){
  if(User_Temp <= Temp_Fever){ //test if user temperature is acceptable
    healthytone();
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("You are healthy!  ");
        lcd.setCursor(0,1);
```

```
        lcd.print("Temp:");
        lcd.print(User_Temp);
        lcd.print("  F");
        Serial.print(F("User is healthy    "));
        Serial.print(User_Temp);
        Serial.println("F");
       }
  else{ //if user temperature is too high, alert user of fever
    fevertone();
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("You have a fever!  ");
        lcd.setCursor(0,1);
        lcd.print("Temp:");
        lcd.print(User_Temp);
        lcd.print("  F");
        Serial.println(F("User is sick"));
        Serial.print(User_Temp);
        Serial.println("F");
       }
   delay(5000);
}

/*****************************************
* Function for when no user interaction occurs
*****************************************/
void gostandby(){
    LCDlines("Going to sleep  ", "                ");
    delay(3000);
    lcd.clear();
    lcd.noBacklight();
    thermal.sleep();
    delay(800);
}

/*****************************************
* Function for turning off system if no user occurs within 5 minutes
*****************************************/
void check5mins(){
  if(lastWakeup < (minute()-5)){
    Serial.println(F("GOING INTO DEAD MODE"));
    LCDlines("Turning off now ", "                ");
    delay(1000);
    lcd.noDisplay();
    thermal.sleep();
    proximity.powerOffProximity();
    while(1);
    LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);
   }
}

/*****************************************
* Function for writing user information to the SD card log
*****************************************/
void printToFile(){
    myFile.print(hour());
    fileprintDigits(minute());
    fileprintDigits(second());
    myFile.print(" ");
    myFile.print(day());
    myFile.print("/");
    myFile.print(month());
```

```
    myFile.print("/");
    myFile.print(year());
    myFile.print(" User Temperature: ");
    myFile.print(User_Temp);
    if(User_Temp>=100.4)
    myFile.print("       USER HAS FEVER"); //print if user has a fever
    myFile.println();
    myFile.println();
    myFile.close(); // close the file
}

/*****************************************
* Main Program
*****************************************/
void loop() {
  unsigned int proxValue = proximity.getProximity(); //search for a user
  check5mins(); //checks if there has been no user input for 5 minutes, will shut down if none

  if(proxValue >10){ //if user in within sensor range, begin measurement loop
    wakeup();

    do{   //while user is within sensor range, stay in measurement loop
        proxValue = proximity.getProximity(); //update user distance
        delay(1000);

          if((proxValue >10)&&(proxValue <80)){   //if user is too far to system, tell user to
move forward
            Serial.println(F("too far"));
            LCDlines("Please Move     ", "Forward         ");
            delay(300);
            }

          else if(proxValue >300){    //if user is too close to system, tell user to move back
            Serial.println(F("too close"));
            LCDlines("Please Move     ", "Back            ");
            }

          else if((proxValue >80)&&(proxValue<300)){    //if user is in measurement range,
take user temperature
            Serial.println(F("in reading zone"));
            LCDlines("Starting Scan...", "Stand Still     ");
            delay(3000);
            getTemp();    //get the user temperature
            testFever();  //determine if user has a fever

            myFile = SD.open("test.txt", FILE_WRITE);
              if (myFile)
                printToFile(); //print user information to file
              else
                Serial.println(F("error opening test.txt")); // if the file didn't open, print
an error:
            delay(300);
            }
      }while(proxValue >10);
      }

  else{
    Serial.println(F("sleep mode"));
    gostandby();  //go into standby mode if there is no user being measured
    delay(1500);
    }
}
```

```
/*****************************************
* Function for playing a pleasant tone when user is healthy
*****************************************/
void healthytone(){
    tone(buzzer,1319); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,1568); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,2637); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,2093); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,2349); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,3136); delay(50); noTone(buzzer);
    delay(130);
}

/*****************************************
* Function for playing a negative tone when user is sick
*****************************************/
void fevertone(){
    tone(buzzer,3136); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,2637); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,3136); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,2349); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,1568); delay(50); noTone(buzzer);
    delay(130);
    tone(buzzer,1319); delay(50); noTone(buzzer);
    delay(130);
}
```