```
/*****************************************************************************************************************
 * Program: ECE_342_Design_Project
 * Author: Hunter Pitzler
 * Date: 2/11/2022
 * Description: When the USB is connected, waits for barcode scanner to detect a barcode. When one is
 *              detected, seven seg display counts down from 3 and IR sensor takes temperature at 0.
 *              Result is displayed on Sevenseg and sent to external database through USB and is
 *              and accompanied by appropriate speaker output. The Device also has the capability to use
 *              a button input to trigger the temperature scan instead of a barcode.
 *****************************************************************************************************************/
#include <math.h>
#include <TM1637Display.h>
#include "SD.h"
#include "TMRpcm.h"
#include "SPI.h"
#include "SoftwareSerial.h"
#include "SparkFun_DE2120_Arduino_Library.h"
#include <avr/sleep.h>

SoftwareSerial softSerial(5, 7); //RX, TX: Connect Arduino pin 5 to scanner TX pin. Connect Arduino pin 7 to scanner RX pin.
TM1637Display display(3, 0);     //CLK, DIO
TMRpcm tmrpcm;                   //Object controlling the speaker
DE2120 scanner;                  //Object controlling the Barcode Scanner

#define SD_ChipSelectPin 4
#define BUFFER_LEN 40
char scanBuffer[BUFFER_LEN];

const float conversion_factor = 5.0/1023.0; //converts ADC reading into a voltage 5V/2^10 bits
//const float temp_const = .02466;
const float temp_const = .003925;          //Calibrated K value
const float exp_const = 2.5;               //Calibrated Delta value

/*****************************************************************************************************************
 * Function: flushRX
 * Description: Flushes/Clears the buffer used to store input from the barcode scanner
 *****************************************************************************************************************/
void flushRX(){                 //Flushes the scanner's buffer
  while(Serial.available()){
    Serial.read();              //Reads '\0's in from USB connection
    delay(1);
  }
}

/*****************************************************************************************************************
 * Function: scan
 * Description: If a barcode is detected the data contained inside the barcode is recorded in the scanner
 *              buffer and the function returns true. Otherwise the function returns false.
 * Inputs:  Barcode read by scanner
 * Outputs: Barcode is stored in scanner buffer if present, and funciton returns appropriate boolean
 *****************************************************************************************************************/
bool scan(){
  flushRX(); //Flush out the buffer
  if(scanner.readBarcode(scanBuffer, BUFFER_LEN)){ //Determines if the scanner is reading a barcode
    for (int i = 0; i < strlen(scanBuffer); i++)   //If so, stores the barcode in the scanner's buffer
    return true; //And returns true
  }
  delay(200);
  return false;                                    //Otherwise returns false
}

/*****************************************************************************************************************
 * Function: play_countdown
 * Description: Causes the seven seg display to countdown from 3 over 3 seconds
 * Outputs: Seven seg display counts down from 3
 *****************************************************************************************************************/
void play_countdown(){
  display.showNumberDec(30); //Display number
  delay(1000);               //Wait a second
  display.showNumberDec(20); //Repeat
  delay(1000);
  display.showNumberDec(10);
  delay(1000);
}

/*****************************************************************************************************************
 * Function: get_voltages
 * Description: Measures the voltages on A1 and A5 and then stores those values
 * in the locations pointed to by the pointers passed in.
 * Inputs: Pointer to location of floats A1 and A5
 * Outputs: Floats A1 and A5 are now the average value of the voltage on pins A1 and A5
 *****************************************************************************************************************/
void get_voltages(float* VA1, float*VA5){
  play_countdown();
  display_hold();
  double VA1_total = 0, VA5_total = 0;
  for(int i=0; i<5000; i++){ //Takes 1000 readings to get average value
     VA1_total += double(analogRead(A1))*conversion_factor;
     VA5_total += double(analogRead(A5))*conversion_factor;
  }
  *VA1 = VA1_total/5000.0; //Stores average of running totals
  *VA5 = VA5_total/5000.0;
}

/*****************************************************************************************************************
 * Function: take_temperature
 * Description: Determines the temperature of the object seen by the IR sensor and returns it
 * Inputs: None
 * Outputs: Temperature of object represented as a float
 *****************************************************************************************************************/
float take_temperature(){
  float temperature, VA1, VA5; //Create necessary variables
  get_voltages(&VA1, &VA5);
  temperature = pow(((VA5/temp_const)+pow(62.8,(4.0-exp_const))*exp(-1.83*((3.3/VA1)-1.0)*(4.0-exp_const))),(1.0/(4.0-exp_const))); //Equation to solve for Tob given VA1 and VA5
  temperature = temperature*1.8 - 32.0; //Converts temperature from Celcius to Fahrenheit
  if(10*(10*temperature-int(10*temperature)) >= 5){ //Effectively rounds up if hundreds place >= 5
    temperature += .1;
  }
  return temperature;
}

/*****************************************************************************************************************
 * Function: send_data
 * Description: Sends barcode number and temperature recorded to database using the serial connection
 * Inputs: Measured temperature and barcode
 * Outputs: Serial write to external database
 *****************************************************************************************************************/
void send_data(float temp){
  char user[45];
  char tenths[1];                          //Create C string variable to send to database
  memset(user, '\0', sizeof(user));        //Configure C string
  strcat(user, scanBuffer);                //Add the barcode scanned from the barcode buffer
  strcat(user, "/");                       //Configure data so external database can read it
  char buff[5];
  memset(buff, '\0', sizeof(buff));
  itoa(temp, buff, 10);                    //Add recorded temperature passed in
  strcat(buff, ".");
  itoa(int(10*(temp-int(temp))), tenths, 10);
  Serial.write(user);                      //Send Serial data over USB connection to database
  Serial.write(buff);
  Serial.write(tenths);
  Serial.write("\r\n");                    //Configure data so external database can read it
}

/*****************************************************************************************************************
 * Function: display_normal
 * Description: Displays the temperature read by the IR sensor on the seven seg display
 * Inputs: Measured temperature
 * Outputs: Temperature displayed on screen
 *****************************************************************************************************************/
void display_normal(float temp){
  display.showNumberDec(temp*10); //Multiply temp by 10 to include 10th's place precision
}

/*****************************************************************************************************************
 * Function: display_fever
 * Description: Displays the temperature read by the IR sensor on the seven seg display. Number blinks
 *              while audio plays.
 * Inputs: Measured temperature
 * Outputs: Temperature displayed on screen
 *****************************************************************************************************************/
void display_fever(float temp){
  for(int i = 0; i<4; i++){
    display.showNumberDecEx(temp*10); //Show temp
    delay(640);                       //Wait
    display.clear();                  //Clear Screen
    delay(640);                       //Wait and repeat
  }
  display.showNumberDec(temp*10);     //Cease blinking when audio ends
}

/*****************************************************************************************************************
 * Function: display_bad
 * Description: Writes BaD to the seven seg display
 * Outputs: Bad is displayed on seven seg display
 *****************************************************************************************************************/
void display_bad(){
  byte rawData[4];
  rawData[0] = 0b01111100; //B
  rawData[1] = 0b01110111; //a
  rawData[2] = 0b01011110; //D
  rawData[3] = 0b00000000; //'\0'
  display.setSegments(rawData, 4, 0); //Write Bad to display
}

/*****************************************************************************************************************
 * Function: display_hold
 * Description: Writes HOLd to the seven seg display
 * Outputs: HOLd is displayed on seven seg display
 *****************************************************************************************************************/
void display_hold(){
  byte rawData[4];
  rawData[0] = 0b01110110; //H
  rawData[1] = 0b00111111; //O
  rawData[2] = 0b00111000; //L
  rawData[3] = 0b01011110; //D
  display.setSegments(rawData, 4, 0); //Write Hold to display
}

/*****************************************************************************************************************
 * Function: button_push
 * Description: Returns true if button is pushed, otherwise returns false
 * Input: State of button
 * Output: Boolean representing state of button
 *****************************************************************************************************************/
bool button_push(){
  return !digitalRead(2);
}

/*****************************************************************************************************************
 * Function: action
 * Description: When scanner or button input is detected, records temperature from IR scanner and displays
 *              it on seven seg along with accompanying audio from speaker. If USB is connected, also
 *              writes barcode and temperature to external database through the USB.
 * Input: bool scanner reports on whether or not the scanner is currently in use
 * Output: Takes temperature and displays it on seven seg, plays appropriate audio, and if applicable
 *         writes serial output to external database.
 *****************************************************************************************************************/
void action(bool USB){
  scanner.lightOff();             //Turn scanning light off
  float temp = take_temperature(); //Get the temperature seen by IR sensor
  display.clear();                //Clear the seven seg display
  tmrpcm.play("Suspense.wav");    //Play dramatic "drum role"
  while(tmrpcm.isPlaying()){}
  analogWrite(9, 0);              //Mute speaker after "drum role" ends

  if(temp < 80.0 || temp > 110.0){ //If an invalid temperature is detected
    tmrpcm.play("Guardian.wav");  //Summons BOTW Guardian
    display_bad();                //Lets user know they are doomed
  }
  else if(temp < 100.4){          //If a normal temperature is detected
    tmrpcm.play("Coin.wav");      //Plays cheerful sound
    display_normal(temp);         //Displays temp on seven seg display
    if(USB){
      send_data(temp);            //Writes data to external database if applicable
    }
  }
  else if(temp >= 100.4){         //If a fever is detected
    tmrpcm.play("RedAlert.wav");  //Sounds Red Alert
    display_fever(temp);          //Displays temp on seven seg display
    if(USB){
      send_data(temp);            //Writes data to external database if applicable
    }
  }

  while(tmrpcm.isPlaying()){}
  analogWrite(9, 0);              //Mute speaker when audio ends
  scanner.lightOn();              // Turn scanning light on
}

/*****************************************************************************************************************
 * Function: setup
 * Description: Initializes program upon start up
 *****************************************************************************************************************/
void setup() {
  Serial.begin(115200);             //Initialize Serial Connection
  pinMode(9, OUTPUT);               //Initializes pins
  pinMode(2, INPUT);
  pinMode(A1, INPUT);
  pinMode(A5, INPUT);
  display.setBrightness(7);         //Sets seven seg display brightness
  tmrpcm.speakerPin = 9;            //Sets speaker pin

  if (!SD.begin(SD_ChipSelectPin)){ //Verifies SD card is connected
    Serial.println("SD fail");
    while(true);
  }
  scanner.begin(softSerial);        //Initializes scanner
  scanner.changeBuzzerTone(1);      //Sets scanner buzzer tone
  scanner.lightOn();                //Turn scanning light on
  scanner.reticleOn();              //Turn scanning laser on
  scanner.enableMotionSense(100);   //Turn on motion sensing mode
}

/*****************************************************************************************************************
 * Function: loop
 * Description: Main loop that runs the program
 *****************************************************************************************************************/
void loop() {
  if (Serial){                      //Checks if USB is connected
    if(scan()){                     //Checks if barcode was scanned
      action(true);                 //Takes action if barcode was scanned
    }
  }
  if(button_push()){                //Checks if button was pushed
    action(false);                  //Takes action if button was pushed
  }
}
```