Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5– No. 2, May 2016 – www.caeaccess.org

Sensor based Hyperthermia Alert Car Application

Maysoon F. Abulkhair
Department of IT
Faculty of Computing and Information Technology
King Abdulaziz University

Amani Aldahiri
Department of IT
Faculty of Computing and Information Technology
King Abdulaziz University

Hanin Alkhatabi
Department of IT
Faculty of Computing and Information Technology
King Abdulaziz University

Hala Alonezi
Department of IT
Faculty of Computing and Information Technology
King Abdulaziz University

Lujain Mulla
Department of IT
Faculty of Computing and Information Technology
King Abdulaziz University

Somia Razzaq
Department of IT
Faculty of Computing and Information Technology
King Abdulaziz University

ABSTRACT
Leaving children unattended in car for few moments, especially in hot atmosphere can cause a catastrophic tragedy to occur. Even young children can die of hyperthermia after being left in hot cars for a period of time. This paper presents HACC (Hyperthermia Alert for Children in Cars) system to help in preventing tragic child death caused by hyperthermia using detection and control system inside car. HACC system has been accomplished by developing a phone application and a surveillance system connected together to monitor temperature and presence of a child inside car. The system starts to measure temperature inside car via temperature sensor. At the same time, it checks constantly the presence of child inside car via motion sensor. When system detects the presence of a child and temperature inside car reaches unsafe limit, it alerts caregiver via smartphone application and allows him/her at the same time to take an action and open windows remotely. If there is no response from caregiver, system itself reacts and windows are opened automatically.

Keywords
Hyperthermia, HACC, Surveillance system, Mobile application, Sensors

1. INTRODUCTION
Nowadays, cars are the most popular transportation that brings convenience and ease for people to perform their daily activities. However, many families have constant fear and suspicions about safety of their children inside cars. Many solutions were made to limit these dangers by preventing children from being injured or getting hurt by using the best car's safety equipment or car's safety application.

Parents around the world must do their intense researches on keeping their children safe inside car and shop for best children car's seats. However, securing the children in proper car seats is not enough to assuring their safety, because there are other dangerous around them that might harm or even kill them.

Hyperthermia (heatstroke) is one of the most common dangers that can affect children lives. It is considered the third cause of children deaths inside cars around the world. It occurs when body is not able to cool itself quickly enough and body temperature rises to a dangerous level. Young children are particularly at risk as their body heats up 3 to 5 times faster than an adult [1].

"Since 1998, until 2012 (14 years) more than 530 children across the United States have died from hyperthermia when unattended in a vehicle [2]." Imagine how many numbers of children have been died by the same phenomenon around the world. Every year, many avoidable deaths occur due to vehicular hyperthermia and rising numbers are worrying child safety. As children are trapped or placed out of sight in back seat of vehicles, the incidence of deaths due to this phenomenon has increased. Preventing this dramatic event of children lives, can be accomplished with a surveillance and detection system implemented inside car to monitor children and protect them from hyperthermia.

For children less than 4 years, heatstroke occurs when their body temperature exceeds 104°F. In addition, the 107°F is considered a lethal body temperature because it damages the body's cells and internal organs shut down [3]. On average, studies have shown that in 10 minutes the temperature in a vehicle raises about 20°F. Within an hour, the temperature jumps around 50°F. So even if the temperature outside vehicle does not feel warm enough to heat it; thus it is better not to leave a child unattended in vehicle because it is not safe [6][4]. Table 1 shows difference of temperature inside a closed vehicle compared to the outside temperature.

Table 1. Difference between Temperature inside and outside the Vehicle

Temperature Inside Vehicle (°F)	Temperature Outside Vehicle (°F)
100	75
120	85
140	100

According to another study, children left in car even for short periods in moderate surrounding temperature (say 21c) are risk of hyperthermia. The internal temperature within closed cars ascends rapidly in first 15 minutes, may be with dispute of variations in the rate of increase depending on vehicle type, color and window tainting. Temperature increases by 1.7-1.9°C per 5 minutes. Within 30 minutes, 80% of temperature is increased and in 60 minutes vehicle have reached peak

temperature regardless whether window is closed or cracked open. [7]

There are two major factors, which make children more delegate than adults to hyperthermia. One is children have more surface area to mass ratio compare to adults and other is children's thermoregulation is less efficient compare to adults. Below the window level, children are well clothed also settled on well-cushioned seat this gives very less area to natural cooling. Therefore child are especially prone to develop hyperthermia when inside car. Thus, vehicular heat stroke tragedies change the lives of parents, families, and communities forever. [8]

2. PROPOSED SYSTEM

Hyperthermia Alert for Children in Cars (HACC) is an interactive safety system provided to monitor the presence of the children who are left behind inside closed, parked vehicle. It helps in preventing tragic child deaths caused by hyperthermia by implementing detection and control system in cars.

HACC system consists of two parts; first car surveillance system which checks temperature constantly and detects presence of a child inside a car. This part is accomplished by installing temperature sensor, motion sensor, GSM shield, Bluetooth chip, Arduino and a buzzer inside the car.

The second part is mobile application that enables user (parents or caregiver) to keep in touch, monitor their child inside the car, and take appropriate reaction to help him/her through smartphone application. This is done by implementing an android mobile application. The application is connected to surveillance system inside a car and receives notification about the car temperature. Moreover, it alerts application's user when the temperature inside the car reaches unsafe limit for a child to be inside it. When there is no response taken, system takes an action and opens car's windows automatically. At the same time, it starts an alarm to grab people attention around the car to get the required help for the child.

3. ACTIVITY DIAGRAM REPRESENTATION

In more detail, Fig 1 represents flow of above proposed system processes. This system starts to measure temperature inside car constantly. If temperature reaches 85°F, then the system starts to check the presence of a child inside the car by detecting his/her motion. If it detects child's motion, system sends a notification to alert parents via phone application. While if it does not detect any child motion inside the car, it returns to measure temperature inside the car. After sending notification, the system allows the user to take an action and open car's windows remotely through the smartphone application. If user presses "Open Windows" button in the application, car's windows will be opened remotely. However if there is no action taken from the smartphone, the system will continue to check the temperature inside car, until it reaches 95°F, the system itself will react and car's windows will be opened automatically. At last, system will be reset and started from the beginning.

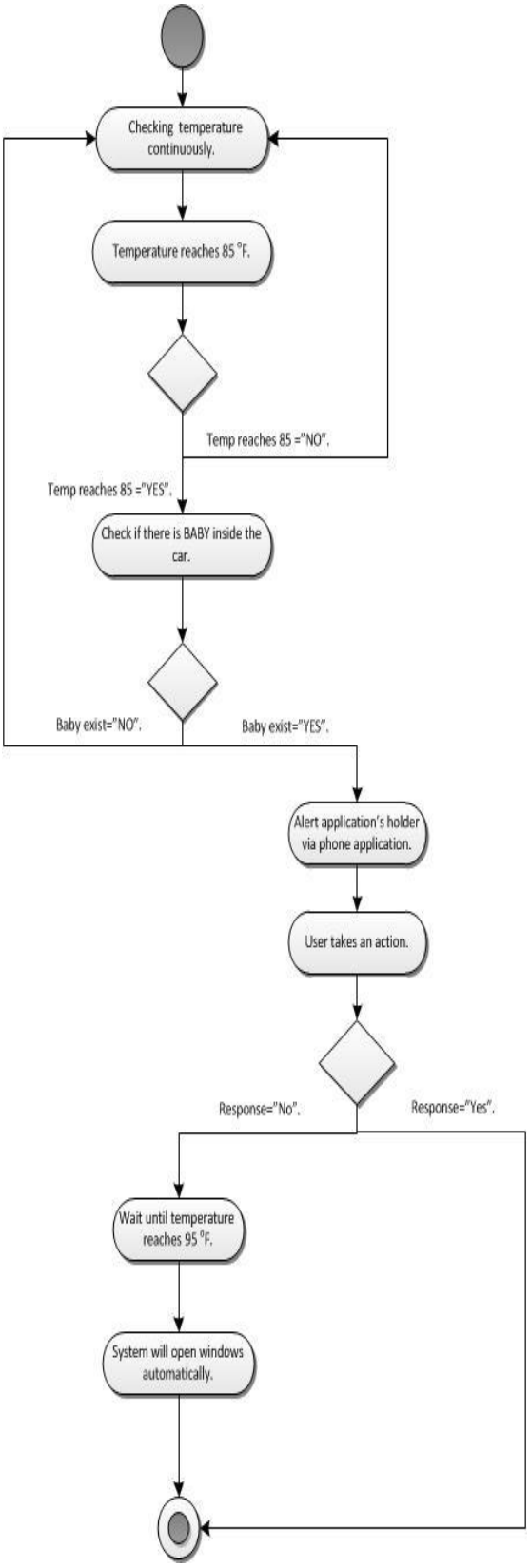

Fig 1: HACC System Activity Diagram

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5– No. 2, May 2016 – www.caeaccess.org

4. EXISTING SOLUTION

Currently, there are variety of technologies utilized to monitor car temperature and to detect the existence of the children in the locked car by alerting the parents or caregivers. These technologies are divided into three main categories [3]:

I. pressure based system: is a connection between pad and weight sensors that detect the presence of the child, such as "smart car seat" presented by [4]. Another example, Car Seat Monitor (Cars-N-Kids) idea is to sense child's presence in child car seat inside car. This was done using a small sensor pad that is placed under the cushion of child car seat. The device is synchronized with mobile phone application that is installed in parent's smartphone. When a child is placed in child car seat, the associated weight activates the device. If the car comes to a stop for more than four seconds, mobile phone vibrates and/or sends a text message to smartphone application. The purpose of this device is to remind parent that child is present before he or she leave the car. [9]

II. Child restraint-based wearing system: designed specifically for children in child restraints, and consisted of two synchronized devices such as the child chest clip and the key fob for parents or caregiver. For example, ChildMinder smart clip system that alerts the caregiver with the alarm attached to the key ring after moving 15 feet from the associated child and within six seconds of the movement.

III. Vehicle-based wearing system detected the existence of the child setting in the rare seats of the vehicle; like backseat minder that can determine the child existence from the number of opening and closing the car door within three seconds.

Another proposed solution is a multi-agent system to monitor the baby seats inside cars. This is also a trustworthy mechanism to alert parents or caregivers whom accidentally leave their children in a baby seat inside an enclosed car [14].

5. EVALUATION

After studying the previous systems, it is observed that the implemented device is just detecting child's presence only in child car seat using weight sensor. Further, the utilization of mobile phone application is the best way to alert parent remotely, easy to use and more practical.

On the other hand, goal of HACC system is to check presence of child anywhere inside the car using motion sensor. Unlike previous systems and technologies, HACC system is designed to work in both situations either child is left unattended intentionally or inadvertently inside a car. Instead of using text message notification, HACC uses an alarm sound notification, emitted from smartphone application to alert parent/caregiver. Finally, HACC system is designed to act automatically to save children from hyperthermia danger even if no action is taken from parent or caregiver. Table 1 illustrates the comparison between the implemented systems for sensing the presence of the child and Hyperthermia Alert for Children in Car (HACC).

Table 2. Comparison between Sensing the Presence of the child Systems and HACC System

Criteria	Sensing the Presence of the child Systems	HACC System
Functionality	It detects child's weight on child car seat only or his/ her movement.	It detects any child's motion inside whole car.
Conditions	It works when child left unattended inadvertently inside car.	It works when child left unattended intentionally or inadvertently inside car.
Compatibility	It works with child car seats only; it is not compatible with all cars.	It is compatible with all types and sizes of cars.
Distance Limit	It works around the car only.	It works remotely.
Response Action	It does not allow parents to take any action remotely to save their child.	It allows parents to take action remotely to save child. Even, it automatically reacts and rescues child, if no response has been taken by parent. Additionally, it connects with emergency organization directly fast rescue.
Language	It is available in English only.	It is available in other languages such as Arabic.

6. HACC IMPLEMENTATION TOOLS

Proper tools are used to implement HACC system components. This section describes implementation tools used for two main components (surveillance system, mobile application) in detail.

6.1 HACC Surveillance System

Software and hardware tools used for this component are described in detail in following sections.

6.1.1 Software Tools

In the surveillance part of HACC component, Arduino programming language is used to develop interactive objects, taking inputs from a variety of sensors and controlling outputs. [11]

6.1.2 Hardware Tools

The main hardware tools that are used in implementing HACC system are:

I. LM35 temperature sensor is used to measure temperature. [12]

II. Parallax PIR motion sensor is used to detect child's presence through his/her motion. [13]

III. JY-MCU Bluetooth Wireless Serial Port Module is

used to send signal to the application when temperature inside the car reaches 85°F. [14]

IV. Piezo Buzzer is used when temperature reaches 95°F to attract people's attention around the car and the windows open automatically.

V. Jumper Wires which are used to connect the sensors to the Arduino board.

VI. White Breadboard is used to put the sensors and connect wires on it.

VII. USB cable is used to connect Arduino software to Arduino.

6.2 HACC Smartphone Application

Software and hardware tools used for this component are described below in detail.

6.2.1 Software Tools

Android Software Development Kit (SDK) is used to develop the application for Android platform. It includes development tools, an emulator and required libraries. Applications are written using Java programming language and run on Dalvik.

6.2.2 Hardware Tools

A smart mobile phone that supports Android operating system.

7. HACC IMPLEMENTATION PROCESS

In HACC system, two Arduino microcontrollers are used. The first Arduino is used to check temperature in Fahrenheit degree constantly via temperature sensor inside car. Moreover, it starts an alarm in car via buzzer to attract people's attention to provide required help for child. The second Arduino is used to detect child's motion via motion sensor. In addition, under certain circumstances, it sends a notification to parent's phone application via Bluetooth chip.

In more detail, master and slave technique is used that allows to setup two or more Arduino boards to share information with each other using I2C synchronous serial protocol. Moreover, parent's smartphone application prompts user to use multiple option to react and sends response back to car application via SMS. This section explains all steps in preparing and installing Arduino Uno and its related used hardware to implement HACC system.

7.1 HACC Surveillance System Implementation

7.1.1 First Arduino Code

First Arduino is set as a Master that is programmed to request and read input. The first Arduino consists of Arduino Uno Boards, White Breadboard, LM35 Temperature Sensor and wires.

Steps:

The following steps are done:

i. Prepared Arduino Uno and white breadboard.

ii. Put LM35 temperature sensor in white breadboard.

iii. Made configuration using three wires which are red, blue and black to connect temperature sensor with Arduino.

iv. Used the 5 Volt, Ground and A0 pins in Arduino Uno.

v. Programmed sensor using Arduino software via USB cable.

vi. Uploaded code to Arduino.

vii. Finally, Arduino is tested through serial monitor.

Figure 2 shows the first Arduino configuration, while Figure 3 describes first Arduino code.

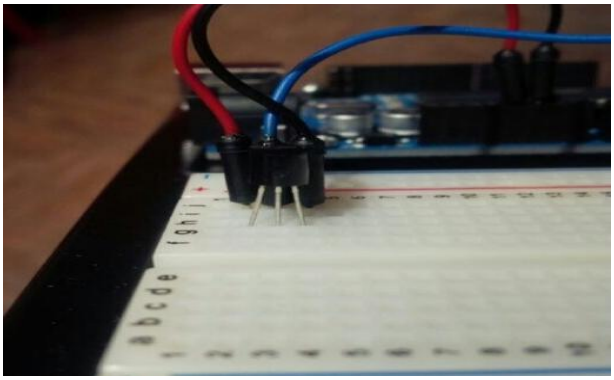

Fig 2: Temperature Sensor Configuration

Code is shown below in figure 3.

```
#include <Wire.h>

const int inPin = 0;
const int outPin = 7
const int tempbefor = 85;
const int tempafter = 95;
const int value;

void setup()
{
  pinMode(outPin, OUTPUT);
  Wire.begin();
  Serial.begin(9600);
}

  //====================

void loop()
{
  int value = analogRead(inPin);

  long celsius =  (value * 500L) /1024;
  long fah = (celsius * 9)/ 5 + 32;

  Serial.println(fah);
  Serial.println(celsius);

     if (fah >= tempbefor && fah < tempafter)
       {
         Wire.requestFrom(4,1);
         while(Wire.available()) {
           char c =Wire.read();
           Serial.println(c);
           digitalWrite(outPin, LOW);
          } }

else if (fah >= tempafter)
   {
     Wire.requestFrom(4,1);
     while(Wire.available())    {
          char c =Wire.read();
          Serial.println(c);
          digitalWrite(outPin, HIGH);
          if(c == 'y') {
                  noTone(13);
                  tone(13, 523, 12000);
                }   }
   } delay(1000);
 }
```

Fig 3: First Arduino Code

7.1.2 Second Arduino Code

Second Arduino is set as a Salve, which waits for a request from first Arduino. After getting a request, it goes to

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5– No. 2, May 2016 – www.caeaccess.org

RequestEvent function (see in Fig 8) to check if there is a motion inside a car.

It consists of Arduino Uno Boards, Parallax PIR Motion Sensor, JY-MCU Bluetooth Wireless Serial Port Module, White Breadboard and Wires.

Steps to Setup Motion Sensor:

Following steps are done:

i. Prepared Arduino Uno and white breadboard.

ii. Put Parallax PIR motion sensor in white breadboard.

iii. Made configuration using three wires which are orange, blue and green to connect motion sensor with Arduino.

iv. Used the 5 Volt, Ground and digital "2" pins in Arduino Uno.

v. Programmed sensor using Arduino software via USB cable.

vi. Uploaded code to Arduino.

vii. Finally, tested Arduino through serial monitor.

Figure 4 below shows setup for the second Arduino with the motion sensor.

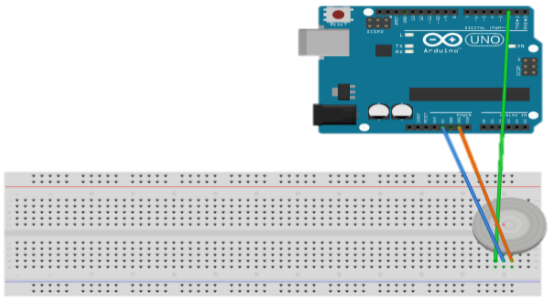

Fig 4: Motion Sensor Configuration

Steps to Setup Bluetooth Chip:

Following steps are done:

i. Prepared Arduino Uno and white breadboard.

ii. Put JY-MCU Bluetooth wireless chip in white breadboard.

iii. Made configuration using four wires which are red, blue, black and green to connect Bluetooth chip with Arduino.

iv. Used the 3.3 Volt, Ground, TX and RX pins in Arduino Uno.

v. Programmed sensor using Arduino software via USB cable.

vi. Uploaded code to Arduino.

vii. Finally, Arduino is tested through serial monitor.

Figure 5 shows the setup for the second Arduino with the Bluetooth chip.

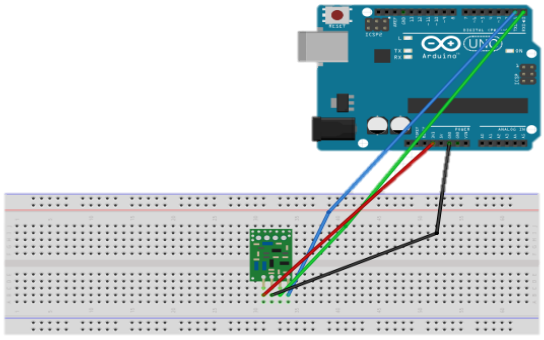

Fig 5: Bluetooth Chip Configuration

While, Figure 6 shows the complete second Arduino configuration.

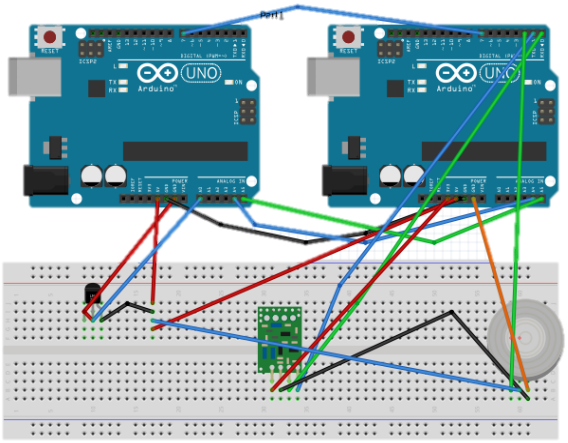

Fig 6: Second Arduino Configuration

Furthermore, the programming code for the connection between both the motion sensor and Bluetooth chip with the Arduino is presented in Figure 7 and Figure 8 respectively.

Steps to Setup Buzzer:

Following steps are done:

i. Prepared Arduino Uno and white breadboard.

ii. Put buzzer in white breadboard.

iii. Made configuration using two wires that are red and black to connect buzzer with Arduino.

iv. Used the Ground and "13" pins in Arduino Uno.

v. Programmed sensor using Arduino software via USB cable.

vi. Uploaded code to Arduino.

vii. Finally, tested Arduino through serial monitor.

Figure 9 shows the setup for the Arduino with buzzer.

```
void RequestEvent()
{
   signal = digitalRead(inputPin);
   temp = digitalRead(inpin);

   if (signal == HIGH) {

     if (pirState == LOW) {

        if (temp == LOW && count == 0) {
             a=1;
             count++;
                   }

        else if(temp == HIGH && count1 == 0) {
             a = 2;
             count1++;
             digitalWrite(outPin, HIGH);
             delay(250000);
             digitalWrite(outPin, LOW);
                   }

      Wire.write("y");
      pirState = HIGH;
            }
         }

    else if (signal == LOW){

       Wire.write("n");
       pirState = LOW;

                 }
               }
```

Fig 7: Second Arduino Code 1

```
#include <Wire.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(0, 1);

char incomingByte;
int inputPin = 2;
int outPin = 13;
int inpin = 7;
int pirState = LOW;
int signal = 0;
float a=0;
float count = 0;
float count1 = 0;
int temp = 0;

//============================

void setup()
{
  pinMode(inputPin, INPUT);
  pinMode(outPin, OUTPUT);
  pinMode(inpin, INPUT);
  Wire.begin(4);
  Wire.onRequest(RequestEvent);
  Serial.begin(9600);
}

//============================

void loop() {
signal = digitalRead(inputPin);
temp = digitalRead(inpin);

  if (signal == HIGH && temp == LOW){
    Serial.println("5");
    if (Serial.available() > 0) {
        incomingByte = Serial.read();

           if(incomingByte == '1') {
              digitalWrite(outPin, HIGH);
              delay(3000);
              digitalWrite(outPin, LOW);
       }
     }
   }
 }

//============================
```

Fig 8: Second Arduino Code 2

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5– No. 2, May 2016 – www.caeaccess.org

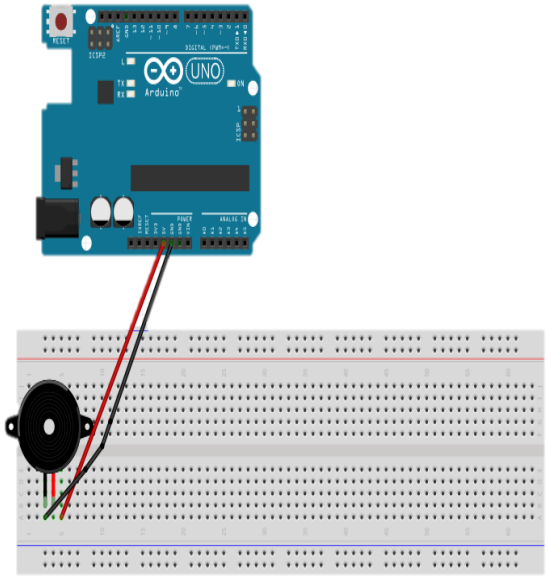

Fig 9: Configuring the Buzzer

The code for programming Arduino with buzzer is presented in Figure 10.

```
if(c == 'y')
    {
        noTone(13);
        tone(13, 523, 12000);
    }
```

Fig 10: Buzzer Connection Code with Arduino

Figure 11 shows the final configuration of the surveillance system included in HACC.

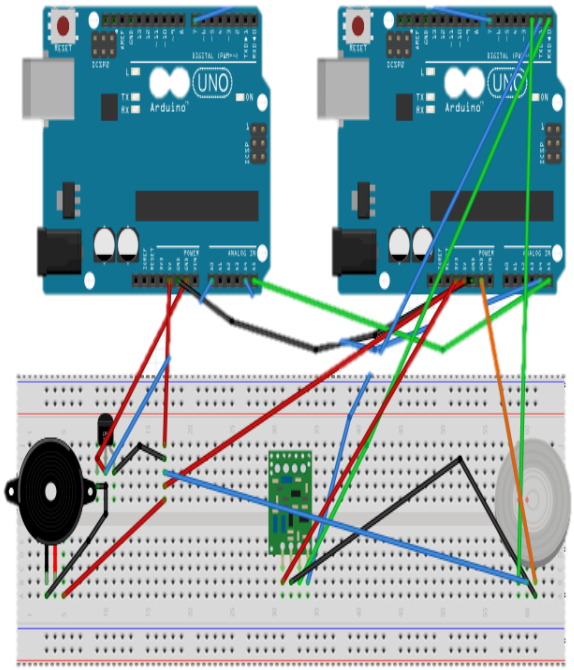

Fig 11: Final Configuration of Surveillance System

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5– No. 2, May 2016 – www.caeaccess.org

```
public class MySmsReceiver extends BroadcastReceiver {

    private static final String SMS_RECEIVED = "android.provider.Telephony.SMS_RECEIVED";
    private String yourNumber = "+966530337212";

    @Override
      public void onReceive(Context context, Intent intent) {
          if (intent.getAction().equals(SMS_RECEIVED)) {
              Log.v("MySMS", intent.getAction());
              Bundle bundle = intent.getExtras();
              if (bundle != null) {
                  Object[] pdus = (Object[]) bundle.get("pdus");
                  final SmsMessage[] messages = new SmsMessage[pdus.length];
                  for (int i = 0; i < pdus.length; i++) {
                      messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                  }
                  if (messages.length > -1) {
                      String no = messages[0].getDisplayOriginatingAddress();
                      Log.v("MySMS", no);
                      if (no != null && no.trim().equals(yourNumber))
                      { Log.v("if",no);
                          PackageManager manager = context.getPackageManager();
                          Intent resultIntent = new Intent();
                          resultIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                          resultIntent.setAction(Intent.ACTION_MAIN);
                          resultIntent.setComponent(new ComponentName(
                                  "kau.it.h.a.c.c",
                                  "kau.it.h.a.c.c.Alert"));
                          ResolveInfo ri = manager.resolveActivity(resultIntent,
                                  Intent.FLAG_ACTIVITY_NEW_TASK);
                          if (ri != null) {
                              context.startActivity(resultIntent);
                          }
                      }
```

Fig 12: Parent/Caregiver's Application Code

```
private void sendSMS(String phoneNumber, String message) {
  PendingIntent pi = PendingIntent.getActivity(this, 0, new Intent(this, MainActivity.class), 0);
  SmsManager sms = SmsManager.getDefault();
  sms.sendTextMessage(phoneNumber, null, message, pi, null);

}
```

Fig 13: Send SMS Automatically Code

7.2 Phone Application

7.2.1 Parent/Caregiver Android Phone application Code

This section represents the programming code for parent/caregiver's smartphone application.

7.2.1.1 SMS Receiver Class

In Figure 12, class "MySmsReceiver" extends BroadcastReceiver base class of android that responds to system-wide broadcast announcements. Many broadcasts originate from the system. For example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. In this code, announcement is the receiving of an SMS and how to handle this announcement is the work of "onReceive()" method.

This class works in the background even when the application is closed.

- The receiver will keep listening until an SMS arrives.
- Upon the arrival of an SMS it will check if it comes from the specified phone number in the variable

"yourNumber" which is the car android application number.

- If it is from "yourNumber", the alert activity will be launched, so that the parent/caregiver can take appropriate reaction.
- Else, it will pass without distracting the user.

7.2.1.2 Send SMS Method

In Figure 13, the "sendSMS" method which has two string parameters the phone number to send the SMS to, and the message to be send. In this method, pending intent is used which is a token that is give to a foreign application (e.g. NotificationManager, AlarmManager, SMS Manager, or other 3rd party applications), which allows the foreign application to use application's permissions to execute a predefined piece of code. Message is sent automatically without opening SMS application or requiring user involvement.

7.2.2 Car Android Application Code

This section represents programming code for car's phone application.

7.2.2.1 SMS Receiver Class

In Figure 14, Class "MySmsReceiver" extends the BroadcastReceiver base class of android that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. In this code, announcement is the receiving of an SMS, how to handle this announcement is the work of "onReceive()" method.

This class works in the background even when the application is closed.

- The receiver will keep listening until an SMS arrives,
- Upon the arrival of an SMS it will check if it comes from the specified phone number in the variable "yourNumber" which is the car android application number.
- If it is from "yourNumber", the Main activity will be launched, so that the to deals with the arduinos via Bluetooth.
- Else it will pass without distracting the user.

7.2.2.2 Send SMS Method:

The "sendSMS" method is used witch have two string parameters the smartphone number to send the sms, and the message to be send. In this method, pending intent is used which is a token that is given to a foreign application (e.g. Notification Manager, Alarm Manager, SMS Manager, or other 3rd party applications). It allows the foreign application to use application's permissions to execute a predefined piece of code. Message is sent automatically without opening the SMS application or requiring user involvement.

```
public class MySMSReceiver extends BroadcastReceiver{

    private static final String SMS_RECEIVED = "android.provider.Telephony.SMS_RECEIVED";
    private String yourNumber = "+966530337212";


@Override
public void onReceive(Context context, Intent intent) {
    if (intent.getAction().equals(SMS_RECEIVED)) {
        Log.v("MySMS", intent.getAction());
        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            Object[] pdus = (Object[]) bundle.get("pdus");
            final SmsMessage[] messages = new SmsMessage[pdus.length];
            for (int i = 0; i < pdus.length; i++) {
                messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
            }
            if (messages.length > -1) {
                String no = messages[0].getDisplayOriginatingAddress();
                Log.v("MySMS", no);
                if (no != null && no.trim().equals(yourNumber))
                { Log.v("if",no);
                PackageManager manager = context.getPackageManager();
                Intent resultIntent = new Intent();
                resultIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                resultIntent.setAction(Intent.ACTION_MAIN);
                resultIntent.putExtra("sms", "1");
                resultIntent.setComponent(new ComponentName(
                        "com.example.bluetooth2",
                        "com.example.bluetooth2.MainActivity"));
                ResolveInfo ri = manager.resolveActivity(resultIntent,
                        Intent.FLAG_ACTIVITY_SINGLE_TOP);
                if (ri != null) {
                    context.startActivity(resultIntent);
                } }
```

Fig 14: Car's Phone Application Code

7.2.2.3 "checkBTState" Method:

This "checkBTState" method checks if the Bluetooth is not supported on the device:

- If it is supported then it checks if it is enabled or it requests to enable it.
- Else, it shows an error message.

7.2.2.4 "createBluetoothSocek" Method:

This "createBluetoothSocek" method it creates a Bluetooth socket connected to the device specified by device ID defined earlier.

7.2.2.5 "Response" Method:

The "Response" method gets the data intent that lunched the application from the MySmsReceiver. It gets the data send via intent and respond to the arduino via Bluetooth writes' method.

7.2.2.6 Thread Class

A thread class is used to get the input, output streams of the connection, read the data send from the Bluetooth via the connection, and pass it to the handler.

8. TESTING

For this HACC system, usability testing is done along with other testing techniques. Usability testing is most important for this system as it is used for user-centered interaction designed system to test product by multiple different users.

For conducting usability testing in HACC system, a test scenario was papered (given in section 8.1) and test was taken by multiple users.

One member of testing team performed as a facilitator. Her work was to give HACC application overview, define usability and purpose of usability testing to participants for HACC system.

Other member worked as test observer who acted like a note taker to record participant's actions, behaviors and comments. Also she observed the participants during test and identified problems and procedural errors. In addition, another member was assigned to respond to participant's questions and requests for assistance.

8.1 Test Scenario

Scenario 1:

Starting from the application home page (pop up screen):

The user need to respond to notification that comes to his/her mobile.

Time started scenario: .....

Time ended scenario: .....

Observer notes:

1. Completed task easily.
2. Completed task with some effort/confusion.
3. Completed task with assistance.
4. Could not complete task

Post-task questions:

How confident are the user successfully completed the task?

1	2	3	4	5
Very Confident	Fairly Confident	Not Confident	Not at all Confident	Didn't complete task

The user asked about his/her impression of the level of ease or difficulty in completing this task on this application?

1	2	3	4	5
Very Easy	Easy	Difficult	Very Difficult	Impossible

Comments

Scenario 2

Starting from the application home page:

The user need to call 997 immediately when the temperature reaches 95°F.15

Time started scenario: .....

Time ended scenario: .....

Observer notes:

1. Completed task easily.
2. Completed task with some effort/confusion.
3. Completed task with assistance.
4. Could not complete task

Post-task questions

How confident are the user successfully completed the task?

1	2	3	4	5
Very Confident	Fairly Confident	Not Confident	Not at all Confident	Didn't complete task

The user asked about his/her impression of the level of ease or difficulty in completing this task on this application?

1	2	3	4	5
Very Easy	Easy	Difficult	Very Difficult	Impossible

Comments

8.2 Analyzing the Study

HACC application is tested with five testers. After the test has completed, it is observed that:

1. 80% of testers they completed the task easily while 20% they completed the task with assistance.

Note: This percentage shows that the application is very usable.

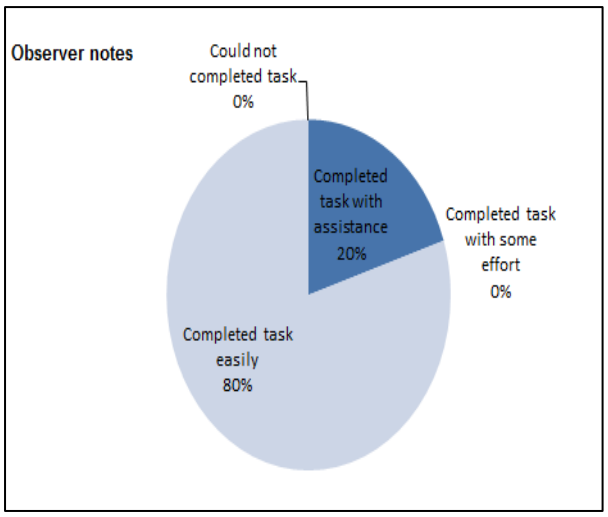

Fig 15: Observer notes for usability testing

2. 100% of the participants say that the application is very easy

Note: This percentage shows that the application is very easy.

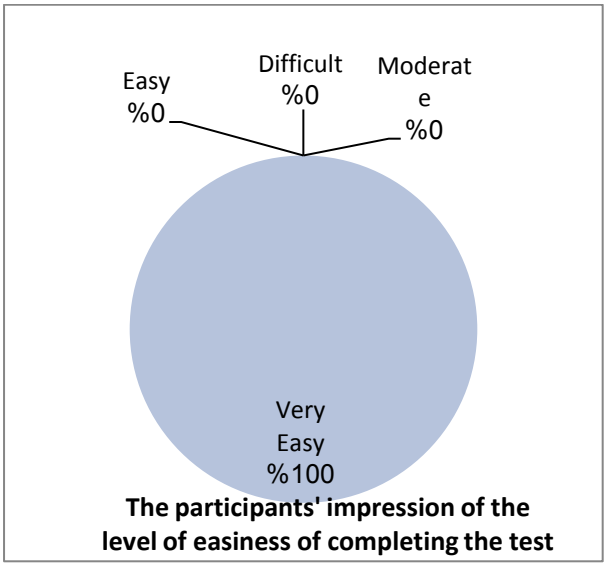

Fig 16: The participants' impression of the level of easiness of completing usability test

80% of the participants say that they are very confident about their ability to complete the task, while 20% were fairly confident.

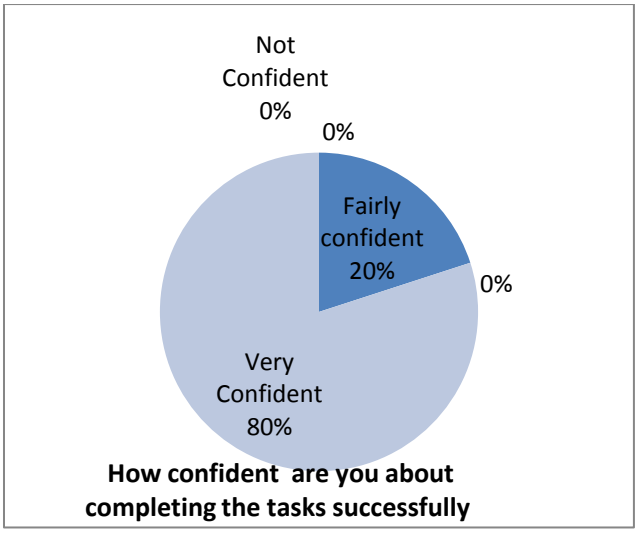

Fig 17: The confident about completing the usability test successfully

9. OVERALL HACC SYSTEM BENEFITS

- Reducing the percentage of children deaths due to this painful incident caused by hyperthermia.
- To maintain the safety of children inside car.
- Works in case of a child left in car intentionally or inadvertently.
- Motivate parents to monitor their children and care about their safety inside car.
- Easy to use.
- Cost affordable.
- The system acts automatically to help child if the parent or caregiver do not provide any help to save him/her.
- Provides a quick instant connection to the emergency center to get help when needed.
- It is compatible with all types of cars.
- It can be implemented in buses.
- The application is available in language other than English such as Arabic.

10. CONCLUSION

In this paper, a simple and low coast system is presented for detecting the unattended child who left intentionally or inadvertently inside car to prevent him/her from hyperthermia. This system is developed based on multi-sensor for both motion and temperature along with smartphone application. In doing so, this system advanced the state of the art by improving the way for preventing children from hyperthermia and enhancing the communication features for more children protection. The enhanced communication features obtained by alerting the parents with a message then if there is no respond then HACC system will automatically act to open the car window and communicate with emergency/servant sector to provide the appropriate help. HACC has been built and tested. The tested results showed that presence of the child

can be detected in an accurate way and the smartphone application is easily evaluated.

In future, the idea of HACC system can be expanded for more levels to cover the issue for minimizing the car temperature. Furthermore, adding a new feature that allows parents/caregivers to watch their children when needed through the smartphone application.

11. REFERENCES

[1] Vittoria Aiello1, Parnian Najafi Borazjani2, Ermanno Battista3, Massimiliano Albanese**,** "Next-Generation Technologies for Preventing Accidental Death of Children Trapped in Parked Vehicles", IEEE IRI, San Francisco, California, USA, August 13-15, 2014.

[2] "Heat Stroke" 10 April. 2013 <http://www.kidsandcars.org/heatstroke.html>

[3] Sumit Garethiya, Himanshu Agrawal, Shilpa Gite, Suresh. V, Amit Kudale, Gaurav Wable, Girishchandra R. Yendargaye, "Affordable System for Alerting, Monitoring and Controlling Heat Stroke inside Vehicles", International Conference on Industrial Instrumentation and Control (IClC), Col/ege o/Engineering Pune, India. May 28-30,2015

[4] Alayna Boland, Michelle Goodman, Jessica Kim, Monica Kruse, Haley Nesmith, and Chelsea Stowell, "Smart Car Seat: sensor design project", https://my.vanderbilt.edu/.../files/.../GrantProposal_Dec2012.docx, 2012

[5] M.Susila, R.Soundarya, S.Vennilavu, B.Karthik, "A CHILD-LEFT-BEHIND WARNING SYSTEM IN VEHICLES ", International Journal of Advanced Technology in Engineering and Science, Volume No.03, Issue No. 03, March 2015.

[6] "Helpful Tips to Keep Your Child Save" http://www.ocfs.state.ny.us/main/publications/Pub5036text.asp

[7] C.P.Shimpi, N.P.Kadam, N.M.Mali "VEHICLE ACTIVE SAFETY SYSTEM: FOR CHILDREN HYPERTHERMIA IN PARKED VEHICLE" Kadam et al., International Journal of Advanced Engineering Research and Studies July-Sept.,2014

[8] Christian Fischer- University of Wuppertal, Thomas Fischer-Delphi Delco electronics Europe GmbH, Brend Tibken-University of Wpuppertal, "The autonomous recognition of left behind passengers in parked vehicle", April 2011.

[9] "Cars-N-Kids" <http://carseatmonitor.com/>

[10] Juan Pablo Soto, Julio Waissman, Pedro Flores-Pérez, and Gilberto Muñoz-Sandoval "Agents That Help to Avoid Hyperthermia in Young Children Left in a Baby Seat Inside an Enclosed Car" © Springer International Publishing Switzerland 2014

[11] "Arduino Uno" <http://arduino.cc/en/Main/arduinoBoardUno>

[12] "LM35 Precision Centigrade Temperature Sensors" <http://www.ti.com/lit/ds/symlink/lm35.pdf>

[13] "PIR Sensor (Rev A)" http://www.jameco.com/1/1/49058-910-28027-pir-sensor-rev-a-detection-range-20-feet-away-motion.html

[14] "Bluetooth to serial HC-06 wireless module" <http://m2.img.dxcdn.com/CDDriver/sku.121326.pdf>