Spring 5-13-2017

Vehicle Occupancy Detection System

Hannah Miller
hmille15@uwyo.edu

Justin Wert
jwert@uwyo.edu

Kelsey Lunberg
klunberg@uwyo.edu

Follow this and additional works at: https://repository.uwyo.edu/honors_theses_16-17

Part of the Electrical and Electronics Commons, Systems and Communications Commons, and the VLSI and Circuits, Embedded and Hardware Systems Commons

Recommended Citation
Miller, Hannah; Wert, Justin; and Lunberg, Kelsey, "Vehicle Occupancy Detection System" (2017). Honors Theses AY 16/17. 27. https://repository.uwyo.edu/honors_theses_16-17/27

This Honors Thesis is brought to you for free and open access by the Undergraduate Honors Theses at Wyoming Scholars Repository. It has been accepted for inclusion in Honors Theses AY 16/17 by an authorized administrator of Wyoming Scholars Repository. For more information, please contact scholcom@uwyo.edu.
Vehicle Occupancy Detection System

EE 4830 – Senior Design II Final Report

Kelsey Lunberg, Hannah Miller, Justin Wert

Advisor: Dr. Jerry Hamann

Attention: Mr. Victor Bershinsky

May 5, 2017
Abstract

Through the course of this senior design project, we were able to create a standalone system implementable by a driver within his or her vehicle for the purpose of occupant detection and notification. The main functions of the developed system include occupant detection via a passive infrared sensor, temperature and humidity sampling, and alerting by means of an external active piezo buzzer in conjunction with a text alert to the parent/guardian of the unattended occupant. We use passive infrared for the purpose of occupant detection due to our original goal of targeting both young children and pets. We felt a void was present in the current market, which caters predominantly to infant children in car seats. It is also imperative to be able to accurately monitor the temperature and humidity conditions in the vehicle for the determination as to whether or not the environment is potentially detrimental to health in the event that an occupant is detected. The heat index is also used to achieve the “real temperature.” This makes use of both the temperature and humidity measurements. The decision was also made to notify both passersby and parents/guardians using a buzzer external to the vehicle and a text message alert which calls for action, respectively. Notifying potential passersby serves as a back-up if the parent or guardian does not take such action in a timely manner. As for the results of our end product, we believe that we achieved both a respectable and functional design. The motion detecting module was packaged separately from the main module. The two systems communicate via an automatic Bluetooth connection on initial power-up. The main module serves to sample temperature and humidity and take necessary alerting actions if the conditions within the vehicle are deemed dangerous for the unattended occupant. The system is powered by both 9V batteries and a simple phone charger from the 12V port in the car. This report will serve as an in depth overview of our system, the components used, schematics and printed circuit board layouts, and other considerations made along the way.
1. **Background**

   Personal vehicles often become traps for unattended occupants throughout the year. Sadly, this can lead to severe health complications or even death if exposed for too long. Although these complications arise primarily in hot climates, it has been demonstrated that they can happen in any environment. Prolonged exposure to an unfamiliar and often dangerous setting without competent human intervention can be detrimental to health. This problem predominantly impacts pets and young children. In an effort to reduce these preventable health complications and possible deaths, we have developed a vehicle occupancy detection system.

   The goal of developing this occupancy detection system was to first determine if an unattended child or pet resided in the vehicle. If detection occurred, the real-time climate was then sampled. If the climate conditions were deemed hazardous or detrimental to health in any way, an alert would then be sent to a parent or guardian via a text message. Main system components include passive infrared detection sensors, temperature and humidity sensors, Bluetooth modules for communication between subsystems, and an SMS module for cellular alert communication. This system process aligns with efforts in reducing as many preventable health complications and deaths due to prolonged exposure to dangerous environments as possible.

   There are currently a few different products on the market that work toward the goal of reducing infant health complications and possible deaths due to prolonged exposure to potentially dangerous climate conditions. According to our research, there are at least two main products on the market that currently have capabilities of detecting if a child is unattended within a vehicle. The first product is called “Sense-A-Life.” Currently it is still in the pre-production stages and is not yet available for purchase. Currently, the organization is accepting distributors and consumers can pre-order the device. There are two parts to the device, one for the driver’s seat and one for the side of the child’s portable car seat. If the driver leaves and the child is left in the vehicle it will inform a list of people
through their cell phone via Sense a Life’s mobile application. Based on our research, we believe that pressure sensors are being utilized to make this system function [14]. Another device that is currently on the market for children’s safety is embedded within the child’s portable car seat itself. Evenflo’s SensorSafe [3] car seat utilizes the On Board Diagnostic system of cars (OBD manufactured past 2008) paired with the chest clip of the car seat. If the ignition of the car is in the off position and the chest clip is still fastened for a sufficient length of time, an alert will sound through the speakers of the vehicle. It works similarly if the chest piece is unclasped while the car is running. Even though both of these products address their respective goals adequately, we felt that we could develop a product which broadened this scope to both pets and children not necessarily constrained by a car seat.

As briefly covered in the abstract, the system that we developed makes use of the popular method of passive infrared detection for the purpose of detecting both pets and small children. The motivation behind this decision was the thought that both young children and pets would show signs of distress through movement in elevated temperature conditions. The secondary sensor module relays information via Bluetooth back to the main module if and when it senses motion within the vehicle. Temperature and humidity is sampled when the passive infrared sensor detects motion and the heat index is calculated based on both measurements. If the calculated heat index is deemed to be detrimental to health in any way, a text message is sent to the parent/guardian of the child or pet for the purpose of alerting them of the possible threat to the unattended occupant in the vehicle. If action is not taken quickly, an external buzzer will sound to alert passersby of the danger the occupant is experiencing.

We began this project with the idea to incorporate multiple passive infrared motion detection sensors within the vehicle for the purpose of reducing false positives. Over the course of the design, we determined that this would be virtually impossible to implement due to the limitations of the hardware in use. Through testing, we learned multiple slave Bluetooth modules could not be connected to one
master Bluetooth module at the same time. The only way that this could be accomplished with the hardware that we are presently using is through a “pinging” process. This process would involve establishing a connection between the master and one slave, a short period in seeing if there is any serial information available, terminating that connection, and finally establishing the Bluetooth connection again between another slave and the master. Obviously, there is still a problem here. Only a single connection is made at any given time and the false positives consideration is still not achievable. Not to mention the fact that this process of “pinging” the individual slave modules would require overwhelming current draw and would, in-turn, cause the operation time of our system to decrease due to elevated draw from the power sources. Therefore, as of now, our project functions via a Bluetooth connection between the two individual modules. Other than this small change in the scope of our project, all of the remaining functionalities remained the same throughout.

2. Functional Description

The technical specifications for our components are as follows:

- **Microcontroller (ATmega328P) [7]**
  - Removable DIP of the Arduino Uno Development Board
    - Will be implemented into our final printed circuit board
  - 14 digital I/O pins
  - Operating Voltage: 5 V
  - -40°C to 85°C operating range
  - Serial and SPI communication supported
- **PIR Sensor (Parallax Sensor) [10]**
  - Operating voltage: 3-6V DC
  - Outputs high on the data pin when motion is detected
  - Selectable detection range (15-30 ft.)
  - 90° field of view
  - Current up to 12 mA at 3V (about 130 µA when idle)
- **Temperature and Humidity Sensor (Adafruit HTU21D-F) [17]**
  - Ideal for heat index calculation capabilities because of the dual measurement
- Uses a capacitive humidity sensor along with a thermistor
- Power, ground, and data pins available
- Operating Voltage: 3-5V
- -30°C to 90°C temperature measurement

**HC-06 Bluetooth Module [19]**
- Transmit and receive capabilities
- Slave serial communication type
  - Must have other type of Bluetooth to pair
- Operating Voltage: 3.1V – 4.2V
- 2.4 GHz digital wireless transceiver
- -25°C to 75°C Operating Range
- Default Baud Rate: 9600bps

**BlueSMiRF Silver (Sparkfun) [18]**
- Uses RN-42 Module
- -40°C to 70°C Operating Range
- Serial Communications: 2400-115200bps
- Operating Voltage: 3.3V – 6V
- FCC Approved Class 2 Bluetooth Radio Modem
- 2.402 to 2.480 GHz Frequency

**FONA GSM Cellular Module [2]**
- SMS text messaging capabilities with multiple frequency bands
  - 850/900/1800/1900 MHz
- -40-85°C operating range
- UART and SPI interface
- 2.8 – 5 V operating range
- Built-in SIM card holder on the reverse side

**Quad-band Cellular Antenna [13]**
- The FONA did not come with this component and it is necessary to complete the module.
- Multi-frequency antenna
System Block Diagram:

Figure 1. System Block Diagram
Looking at Figure 2 above, it is clear that loops will be utilized throughout the code. For testing purposes, these loops were removed in order to give appropriate feedback in a shorter amount of time. In actual implementation, the count loops described above will be utilized to prevent the user from
being bombarded with text messages and giving the parent/guardian/owner some time to address the dangerous situation.

**Schematics:**

![Figure 3. Slave Schematic](image)
Figure 4. Master Schematic

**Functional Description of Systems:**

In order to create a system that is safe and easily installed, wireless communication is necessary. The Passive Infrared Sensor on the Slave Module communicate wirelessly with the main microcontroller module via Bluetooth. To accomplish this, we utilized an additional microcontroller for the sensor that allows the Bluetooth on the Slave Module to transmit to the main module Bluetooth which then transmits the data to the microcontroller. The Slave Module utilizes a Bluetooth device called HC-06. This is a “slave only” module, and requires some other form of Bluetooth module or computer in order to pair. The Master Module utilizes a BlueSMiRF silver Bluetooth Module. This module is able to pair directly with the HC-06 on the Slave Module by setting up auto-pairing, which then searches for the particular address. In future implementation, multiple PIR sensors will likely be necessary. These
modules are not able to connect multiple Slaves to one Master. Therefore, it is pertinent that we are able to “loop” through addresses of the Bluetooth slaves and pair them to the Master. This can be accomplished through code rather than setting up direct auto-pairing.

Detecting presence within the system requires some type of sensor. While motion, pressure, and infrared sensors were all considered, Passive Infrared detection was ultimately chosen to detect an occupant within the vehicle. This sensor uses a combination of both motion and infrared detection. When motion occurs and the heat gradient is changed the sensor goes high, indicating that an occupant within the vehicle has been detected. The passive infrared sensors have a radius ranging from 90° to 120° depending on the sensor. In the future by utilizing an array of multiple sensors, we will able reduce the likelihood of any false positive as well as further span the vehicle to avoid any “dead space” that the PIR isn’t able to pick up. Throughout our initial testing and proof of concept we were able to get a single PIR working and in the future would like to implement multiple slave modules, each with a PIR and appropriate Bluetooth.

To determine the climate conditions of the car, a temperature and humidity sensor was used. Using the heat index calculations as given by the National Weather Service, the car conditions are accurately determined. In order to correctly use the heat index, it is important to have an accurate humidity reading as well as temperature. Using the Adafruit temperature and humidity sensor (HTU21D-F), a single sensor is able to accurately take both temperature and humidity measurements. The temperature range is from -40°F to 257°F with an accuracy of ±1°C (approx. 1.2°F). The operating range for relative humidity is 5-95% with an accuracy of ±1%. Operation is possible outside of these ranges with humidity, the accuracy just decreases a bit.

It is also desired to alert both the guardians/caretakers and pedestrians in the surrounding areas of the vehicle of the unattended child or pet. As referenced above, we made use of a stand-alone system to determine occupancy within the vehicle. This module consisted of the following components:
a microcontroller, battery, Bluetooth slave device, and a PIR sensor. Based on the data that the PIR sensor takes in, the slave device relays a data packet to the master Bluetooth module on the main controller. The main controller will then take this information, process it, and decide whether or not an alert is needed. This alert module consisted of a GSM/SMS module that sends a text to the guardian of the unattended occupant, prompting them to return to the vehicle immediately. In order to alert nearby pedestrians of the impending danger, a piezo-electric buzzer was added to the Master Module. When the dangerous climate conditions are met, a text message is sent and the buzzer goes off. In future implementation, a voice command notifying nearby people of the danger will be utilized, as the buzzer might be seen as a deterrent to nearby pedestrians.

3. **Results**

Overall, each system was empirically tested, both individually, and again once the entire system was complete. Empirical testing was conducted on the temperature and humidity sensor, the communication system (both Bluetooth and SMS Texting), as well as the PIR motion sensor.

The SMS text messaging model was tested in Cheyenne, Wyoming, both separately and with the full Master module. A SMS text message was able to reach Minneapolis, Minnesota from Cheyenne Wyoming, indicating that this module will be able to reach people across the country if necessary. Due to the fact that we utilized a 2G network, we were confined to a very limited area for working cellular service. The furthest from Cheyenne while still in Wyoming we were able to still send text messages was at Interstate 80 mile marker 348. This is a little more than 10 miles outside of Cheyenne city limits.

To test the Bluetooth module, the HC-06 was tested by connecting to a Bluetooth compatible computer while connected to an Arduino and an additional computer. The protocols that were being sent were also tested using a Logic Analyzer, as there were significant technical difficulties encountered. The HC-06 Bluetooth module was able to transmit and receive data from the computer via the Serial Monitor. The BlueSMiRF was tested in a similar fashion, using two computers – one with Bluetooth
capability and one with the BlueSMiRF connected to it to test communication between the Serial Monitor. Once the system was competed, we were then able to connect the two Bluetooth modules together via an auto-pair setting from the BlueSMiRF module. The HC-06 relayed information to the BlueSMiRF when the PIR went high on the Slave Module indicating that an occupant was detected within the vehicle. Once this information was passed to the Master Module, the appropriate sampling of temperature and humidity then took place. The connection was tested at varying distances, and we were able to maintain connection at approximately 25 feet away. This was an indication that the distance within a vehicle was more than adequate to maintain connection.

The temperature and humidity sensor was tested individually by observing the change in humidity and temperature when introduced to new climates via the Serial Monitor. When the final module was complete, the Master Module was placed in a freezer and a temperature of 20°F was obtained for the entire Master Module. Using a hair dryer we were also able to bring the entire system to 117°F and still remain functional. This seems to meet our range for desired temperature goals. If we were in a warmer climate, we would have been able to test higher temperatures.

The PIR was tested individually while on a breadboard. We were able to test the PIR at varying distances to test the sensitivity of the sensor. After iterative testing, it appears that the PIR works for up to around 25 feet with an accuracy of approximately 90%. It seems that the PIR had the best accuracy with large motions as opposed to smaller less-noticeable motions at far distances. As the distance between the motion and the PIR decreased, the sensor became more accurate with smaller motions. The PIR is also able to work in both daylight and in some darkness. When on the roof of a car’s interior, the PIR should not receive direct exposure to sunlight. However, it still worked when exposed to sunlight; the accuracy was simply decreased. Within a vehicle the shorter distance between the occupant and the PIR will be useful, as dramatic movements are not required to trigger the PIR. The range of view of the PIR sensor proved to not require any additional limitation of the sensor to prevent
any false positives. Measurements were taken in two vehicles and it was determined that the field of view from the roof of the vehicle ranged from 111.7° to 120.717°. Once the Slave Module was constructed we were then able to test the module within a vehicle.

Our final product was tested within a vehicle and our design specifications were met. Following our design contract a comprehensive system was created that was able to detect if an unattended occupant was within the vehicle, determine if the interior climate of the car is unsafe for sustained life, and alert the driver and anyone near the vehicle of the unattended occupant. A proper humidity and temperature range was obtained to allow for accurate Heat Index Calculations. SMS Texting was utilized to alert any parent/guardian/owner of the danger, and an additional alert was created to bring attention to the vehicle informing nearby pedestrians of the unattended occupant. The system utilized batteries and the car’s existing 12V power source to provide the system.

Currently the design is a fully functioning module. However, to make it a marketable product, the size of each module would likely need to decrease. It would seem that in the future if we were to market this product, we would want to avoid using any external modules (SMS Texting, Temperature Sensors), and instead incorporate it all into a single PCB. This would allow for a smaller size. Utilizing a 3G cell phone network with GPS capabilities would have been more cost prohibitive, but would have allowed for better cell phone service as well as the ability to relay GPS coordinates to law enforcement. Additional Slave Modules would improve the field of view of the overall system as well as avoid any false positives.

4. Other Considerations

Throughout the design and implementation of this project, we discovered some considerations that we had not foreseen in the beginning. Having a final product now, we can identify some things that we would have done differently and consider aspects that we had not thought of initially.
**Project costs:**

The overall cost for the components placed onto the PCBs and the packaging came to a total of about $175. Throughout the design process there were times where we ordered the wrong component and times where we had to upgrade the components for the functionalities that we wanted. Even with considering these setbacks, the cost wouldn’t have been significantly higher if it were not for the cost of shipping. It can be estimated that we, through the department funding and individual contributions from each of us, spent on the order of anywhere from $300 to $400. It can be said that half of the money spent can be accounted for the shipping and handling of the components. If we were to do this design again we would try to order all component from the least number of manufacturers as well as limited number of orders. Below is a table of project costs.

<table>
<thead>
<tr>
<th>Part</th>
<th>Purpose</th>
<th>Units</th>
<th>Cost/Unit</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTU21D-F</td>
<td>Temp/Humidity Sensor</td>
<td>1</td>
<td>$14.95</td>
<td>$14.95</td>
</tr>
<tr>
<td>ATMega328P</td>
<td>Communication</td>
<td>2</td>
<td>$4.54</td>
<td>$9.08</td>
</tr>
<tr>
<td>FONA Module</td>
<td>SMS Texting</td>
<td>1</td>
<td>$39.95</td>
<td>$39.95</td>
</tr>
<tr>
<td>HC-06</td>
<td>Wireless Communication</td>
<td>1</td>
<td>$8.06</td>
<td>$8.06</td>
</tr>
<tr>
<td>Quad-Band Antenna</td>
<td>SMS Texting</td>
<td>1</td>
<td>$2.95</td>
<td>$2.95</td>
</tr>
<tr>
<td>Lithium Polymer Battery</td>
<td>Power</td>
<td>1</td>
<td>$9.95</td>
<td>$9.95</td>
</tr>
<tr>
<td>USB to Serial breakout</td>
<td>Power</td>
<td>1</td>
<td>$14.95</td>
<td>$14.95</td>
</tr>
<tr>
<td>LCD screen w/ I²C converter</td>
<td>Display</td>
<td>1</td>
<td>$9.99</td>
<td>$9.99</td>
</tr>
<tr>
<td>Other Components</td>
<td>Misc.</td>
<td>x</td>
<td>$10</td>
<td>$10</td>
</tr>
<tr>
<td>Packaging - 1</td>
<td></td>
<td>1</td>
<td>$2</td>
<td>$2</td>
</tr>
<tr>
<td>Ting</td>
<td>SMS Texting</td>
<td>1</td>
<td>-$9</td>
<td>$9</td>
</tr>
<tr>
<td>Ting Subscription</td>
<td>SMS Texting</td>
<td>1</td>
<td>-$9</td>
<td>$9</td>
</tr>
<tr>
<td>Packaging - 2</td>
<td></td>
<td>1</td>
<td>$22</td>
<td>$22</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td></td>
<td><strong>$174.87</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Final Cost Tabulation

**Ethical considerations:**

Something that we would need to be cautious about is the wording of our claims of this design’s ability. This is designed to be a tool to help prevent the occurrences of children and pets being left within the vehicle. Ethically speaking, we would not be able to say that it solves the problem of
hyperthermia within vehicles. In general, any wording of “life-saving” needs to be avoided. This is a tool meant to help mitigate these problems, but it is not a definite solution.

**Social considerations:**

The goal of this project was intended for the greater good. There are some additional sensors that could potentially be added to this design so that it can be more dependable for a wider range of scenarios. For example, if a child or pet were to fall asleep in the vehicle and remain motionless for an extended period of time, our design would not be able to detect the occupant. Utilizing techniques such as infrasound could theoretically detect respiration and would serve as another input to our system. The sociological implications and benefits of the potential mitigation of heat-related illness or death would be hard to deny.

**Political:**

There is a possibility that the government could make this type of design mandatory within vehicles. If this were to happen, improvements to the design would need to be made surrounding any regulations enacted. For instance, it might be required for the power to be dependent on a supply that the user would not have to change and the system might need to have a “user-friendly” way for people to enter their information for the cell phone aspect. If these regulations were to happen, our system would likely serve as an aid in this situation as it has the potential for retrofitting and integration within a wide range of vehicles.

**Health and Safety:**

This project was geared toward the health and safety of car occupants. Given the information the National Weather service provided, hyperthermia and other heat-related injury or illness are a problem in the United States. As stated earlier, more/different types of sensors could be implemented into our design to make this a more reliable product in a situation where the occupant is no longer capable of motion.
**Aesthetics:**

The packaging for both modules turned out to be very nice for prototyping. If this design were to be implemented within a vehicle we believe that the size of each module can be condensed so that it does not take up so much room and would be less conspicuous when integrated into a vehicle. Ideally, this product would have two options: one for retrofitting into a vehicle and one for factory installation in new car models.

**Manufacturability:**

We believe that with some improvements, this design can be easily manufactured. We also think that as more of the modules are made, the more the cost could go down for making the system. To improve manufacturing costs and size, the individual modules would need to be condensed into one singular PCB per module. Rather than using external devices such as the Adafruit FONA, the PCB will be comprised of all necessary sensors and devices for a fully-functioning system.

**Sustainability:**

The main sustainability issue of this system is the two 9 volt batteries. This could be solved using rechargeable batteries. Another improvement could be implementing a monitoring system for the level of charge of the batteries. Once the batteries get discharged to a certain level, the system could send a notification to the user when they need to be replaced or recharged. Another beneficial upgrade to reduce power consumption is an on/off switch or button. This would exist so the user can turn the system off instead of needing to plug and unplug the system each time they get out of the car.

**Appropriate governing standards:**

If this system were to be implemented into a vehicle via a manufacturer, there would no doubt be some regulation and guidelines as to how our system must function. If governing standards are created for health and safety regulations within a vehicle for car manufacturers, any such standard must be followed. Currently, no such regulations exist. Utilizing Bluetooth technology as well as SMS texting in
the manner in which we are might involve FCC regulations if we were to fully market this device. As of right now, this is a proof of concept that demonstrates that a product similar to ours could be made. A device such as this could lead to the creation of these kinds of regulations, which would be beneficial to the general public.
Appendix

PCB Layouts

Figure 5. Printed Slave Module

Figure 6. Slave Module (Improved Version)
One can assume that the PCB circuit in Figure 6 was built after the fact the original PCB was printed. This assumption would be correct. We learned that we could have placed the HC-06 Bluetooth module parallel with the PCB and this would have made the need for the slot in the lid of the enclosure shown in Figure 8 unnecessary.
Packaging Details

In looking at the above two packages, one can see that the slave module is packaged in a modified version of what was originally purposed as a crayon box. The master module is made of Plexiglas and held together by Loctite adhesive. Various nylon screws, standoffs, and nuts were used in both modules for the purpose of locking the PCBs in place, the PIR and LCD in place, and for creating a functional lid for the main module.
Code Listing 1: Master Module with Heat Index Calculation

This example sketch was written by Jim Lindblom for SparkFun Electronics on February 26, 2013 and is released into the public domain.

This example sketch converts an RN-42 Bluetooth module to communicate at 9600 bps (from 115200), and passes any serial data between Serial Monitor and Bluetooth module.

*******************************************************************************
This is an example for our Adafruit FONA Cellular Module

Designed specifically to work with the Adafruit FONA
 ---> http://www.adafruit.com/products/1948
 ---> http://www.adafruit.com/products/2542

These cellular modules use TTL Serial to communicate, 2 pins are required to interface
Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.
BSD license; all text above must be included in any redistribution
*******************************************************************************

/*
THIS CODE IS STILL IN PROGRESS!

Open up the serial console on the Arduino at 115200 baud to interact with FONA

Note that if you need to set a GPRS APN, username, and password scroll down to
the commented section below at the end of the setup() function.
*/

// This code was utilized by us and modified so that we were able to have a fully functioning system

#include <SoftwareSerial.h>
#include "Adafruit_FONA.h"
#include <Wire.h>
#include "Adafruit_HTU21DF.h"
#include <LiquidCrystal_I2C.h>
#include <LCD.h>

#define FONA_RX 2
#define FONA_TX 3
#define FONA_RST 4

SoftwareSerial FONA_TEXT(FONA_TX, FONA_RX);

// this is a large buffer for replies
char replybuffer[255];

// We default to using software serial. If you want to use hardware serial
// (because softserial isn't supported) comment out the following three lines
// and uncomment the HardwareSerial line
// #include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;

// Use this for FONA 800 and 808s
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
// Use this one for FONA 3G
// Adafruit_FONA_3G fona = Adafruit_FONA_3G(FONA_RST);

uint8_t readline(char *buff, uint8_t maxbuff, uint16_t timeout = 0);

uint8_t type;

int bluetoothTx = 10;  // TX-O pin of bluetooth mate, Arduino D2 This corresponds to the "Soft"
// tx/rx on the arduino
int bluetoothRx = 11;  // RX-I pin of bluetooth mate, Arduino D3

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

char array1[] = "Hello!";
Adafruit_HTU21DF htu = Adafruit_HTU21DF();
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7);
float temp;
float tempF;
float Hum;
char bt;
float HI;
float Adj0;
float Adj1;
float HI00;
float calcHI;
uint16_t vbat;

char CMD[4] = {'$','$','$','$','\0'};
char Data[2] = {'D','\0'};
char Data1[1] = {'D'};
char Pair[15] = {'C',',','2','0','1','6','0','9','1','6','7','6','9','5','\0'};

char Hannah[21] = {'3','0','7','3','7','1','7','1','6','6','1'};
char Kelsey[21] = {'3','0','7','6','9','9','5','7','4','1'};
char Justin[21] = {'9','7','0','3','2','4','1','0','5','5'};
char occupantmessage[14] = {"Occupant detected within vehicle. Dangerous heat conditions present. Immediate action required"};
char occupantmessageDanger[141] = {"Occupant detected within vehicle. Extremely dangerous conditions present. Immediate action required"};
char batterymessage[141] = {"Low battery for text messaging device. Charge immediately");
int Buzzer = 12;

void setup()
{
    //Initialize the FONA (text message) Device
    while (!Serial);
    Serial.begin(115200);
    Serial.println(F("FONA basic test");
    Serial.println(F("Initializing....(May take 3 seconds)"));
    fonaSerial->begin(4800);
    if (! fona.begin(*fonaSerial)){
        Serial.println(F("Couldnt find FONA");
        while (1);
    }
    type = fona.type();
    Serial.println(F("FONA is OK");
}

//Initialize the Serial communicator for the bluetooth
Serial.begin(9600);  // Begin the serial monitor at 9600bps
bluetooth.begin(115200);  // The Bluetooth Mate defaults to 115200bps
bluetooth.print("$");  // Print three times individually
bluetooth.print("$");
bluetooth.print("$");  // Enter command mode
delay(100);  // Short delay, wait for the Mate to send back CMD
bluetooth.println("U,9600,M");  // Temporarily Change the baudrate to 9600, no parity
// 115200 can be too fast at times for NewSoftSerial to relay the data reliably
bluetooth.begin(9600);  // Start bluetooth serial at 9600
//Bluetooth is set to Auto-Pair, no need to hardcode pairing

//Initialize the stuff for the temp and hum. sensor:
Serial.println("HTU11D-F test");
pinMode(LED_BUILTIN, OUTPUT);
lcd.setBacklightPin(3, POSITIVE);
lcd.setBacklight(LOW);
lcd.begin(16,2);
lcd.clear();
lcd.print("I am working");
if (!htu.begin()){
    Serial.println("Couldnt find sensor!");
    while (1):
}

pinMode(Buzzer,OUTPUT);
printMenu();
}

void printMenu(void) {
    Serial.begin(115200);
Serial.println(F(""));  
Serial.println(F("[b] read the Battery V and % charged"));  
Serial.println(F("[s] Send SMS"));  
Serial.println(F("[n] Network Status"));  
}

void loop()
{
  if (bluetooth.available())    // If the bluetooth sent any characters
  {
    // Send any characters the bluetooth prints to the serial monitor
    Serial.print((char)bluetooth.read());
  }
  if (Serial.available())      // If stuff was typed in the serial monitor
  {
    // Send any characters the Serial monitor prints to the bluetooth
    bluetooth.print((char)Serial.read());
  }

  // Want to set the character 'bt' to be the serial read from the Bluetooth
  bt = bluetooth.read();      // Has to be bluetooth.read() not the Serial.read()
  // For now:
  if (bt == 'P')              // Indicates that an occupant has been detected
  {
    tempF = (1.8)*htu.readTemperature()+32;
    Hum = hru.readHumidity();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Temp (F): ");
    lcd.setCursor(11,0);
    lcd.print(tempF);
    lcd.setCursor(0,1);
    lcd.print("Humd (%) : ");
    lcd.setCursor(11,1);
    lcd.print(Hum);
  }

  // Calculate the Heat Index:
  HI = -42.379+(2.04901523*tempF)+(10.14333127*Hum)-(0.22475541*tempF*Hum) +
      (.00683783*tempF*tempF) - (.05481717*Hum*Hum) + (.00122874*tempF*tempF*Hum) +
      (.00085282*tempF*Hum*Hum) - (0.00000199*tempF*tempF*Hum*Hum);
  // Adjusting if RH is less than 13% and temperature between 80 and 112 degrees F, this is subtracted
  Adj0 = ((13-Hum)*(0.25))*((13-sqrt(abs(tempF-95)))*(0.05882353)));
  // Adjusting if RH is greater than 85% and temp is between 80 and 87. Added on
  Adj1 = ((Hum-85)*(0.1))*(87-tempF)*(.2));
  // If temperature below 80 degrees F:
  HI80 = 0.5*(tempF+61+(tempF-68)*1.2)+(Hum*0.094);
  if (((tempF>80) || (tempF<112)) && (Hum<13))
  {
    calcHI = HI - Adj0;
  }
}
else if((Hum > 35) && ((tempF > 80) && (tempF < 87)))
{
    calcHI = HI + Adj0;
}
else if(tempF < 80)
{
    calcHI = HI80;
}
else
{
    calcHI = HI;
}

if((calcHI >= 80) && (calcHI <= 102))
{
    Serial.begin(9600);
    while(!Serial);

    FONA_TEXT.begin(9600);
    delay(1000);

    Serial.println("Setup complete");
    Serial.println("Sending SMS....");

    FONA_TEXT.write("AT+CMGF=1\r\n");
    delay(1000);

    FONA_TEXT.write("AT+CMGS="3076895741\r\n");
    delay(1000);

    FONA_TEXT.write(occupantmessage);
    delay(1000);

    FONA_TEXT.write((char)26);
    delay(1000);

    Serial.println("SMS sent!");
    digitalWrite(Buzzer,HIGH);
    delay(1000);
    digitalWrite(Buzzer,LOW);
    delay(500);
    digitalWrite(Buzzer,HIGH);
    delay(1000);
    digitalWrite(Buzzer,LOW);
    delay(20000);
}

if(calcHI < 102)
{

    Serial.begin(9600);
    while(!Serial);

    FONA_TEXT.begin(9600);

delay(1000);
Serial.println("Setup complete");
Serial.println("Sending SMS...");
FONA_TEXT.write("AT+CMGF=1\r\n");
delay(1000);
FONA_TEXT.write("AT+CMGS="3076895741"\r\n");
delay(1000);
FONA_TEXT.write(occupantmessageDanger);
delay(1000);
FONA_TEXT.write((char)26);
delay(1000);
Serial.println("SMS sent!");
digitalWrite(Buzzer,HIGH);
delay(1000);
digitalWrite(Buzzer,LOW);
delay(500);
digitalWrite(Buzzer,HIGH);
delay(1000);
digitalWrite(Buzzer,LOW);
delay(20000);
}

if(fona.getBattPercent(vbat) < 50)
{
Serial.begin(9600);
while(!Serial);
FONA_TEXT.begin(9600);
delay(1000);
Serial.println("Setup complete");
Serial.println("Sending SMS...");
FONA_TEXT.write("AT+CMGF=1\r\n");
delay(1000);
FONA_TEXT.write("AT+CMGS="3076895741"\r\n");
delay(1000);
FONA_TEXT.write(batterymessage);
delay(1000);
FONA_TEXT.write((char)26);
delay(1000);
Serial.println("SMS sent!");
}
void flushSerial() {
    while (Serial.available())
        Serial.read();
}

color readBlocking() {
    while (!Serial.available());
    return Serial.read();
}

uint16_t readnumber() {
    uint16_t x = 0;
    char c;
    while (!isdigit(c = readBlocking())) {
        //Serial.print(c);
    }
    Serial.print(c);
    x = c - '0';
    while (isdigit(c = readBlocking())) {
        Serial.print(c);
        x *= 10;
        x += c - '0';
    }
    return x;
}

uint8_t readline(char *buff, uint8_t maxbuff, uint16_t timeout) {
    uint16_t buffIdx = 0;
    boolean timeoutvalid = true;
    if (timeout == 0) timeoutvalid = false;

    while (true) {
        if (buffIdx > maxbuff) {
            //Serial.println(F("SPACE"));
            break;
        }

        while (Serial.available()) {
            char c = Serial.read();

            //Serial.print(c, HEX); Serial.print("#"); Serial.println(c);

            if (c == '\r') continue;
            if (c == 0xA) {
                if (buffIdx == 0) // the first 0x0A is ignored
                    continue;
                timeout = 0; // the second 0x0A is the end of the line
                timeoutvalid = true;
                break;
            }
            buff[buffIdx] = c;
            buffIdx++;
        }
    }
}
if (timeoutvalid == timeout -- 0) {
    //Serial.println(F("TIMEOUT"));
    break;
}

delay(1);

buff[bufidx] = 0; // null term
return buffidx;
Code Listing 2: Slave Module with PIR Sensor

// by Simon Monk
// www.elegoo.com
// 2016-06-13

// Generic "Blink" sketch
// Utilized by VOD Squad for PIR and Bluetooth Implementation

int ledPin = 5;
int buttonApin = 9;
int buttonBpin = 8;
byte ledManual = 0;

// These might not be necessary, but including it just to be sure:
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10,11);
char myString[] = "Person";
void setup()
{
    pinMode(ledPin, OUTPUT);
    pinMode(buttonApin, INPUT);
    //Serial.begin(9600);
    //mySerial.begin(9600);
}

void loop()
{
    if (digitalRead(buttonApin) == HIGH)
    {
        digitalWrite(ledPin, HIGH);
        //Serial.print((char)mySerial.read());
        //Serial.print("Person");
        Serial.begin(9600);
        mySerial.begin(9600);
        mySerial.print('P');  // This was the command that worked -- was originally "P", needed to
        // change it to 'P', should be a character, not a string
        Serial.write("P");

        delay(500);
        digitalWrite(ledPin, LOW);
    }
    else
    {
        digitalWrite(ledPin, LOW);
    }
}
References


    https://learn.adafruit.com/adafruit-htu21d-f-temperature-humidity-sensor/overview

    https://www.sparkfun.com/products/12577

    https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf