Vehicle Child Seat Alert System

Christopher Atkinson   
College of Engineering and Computer Science  
University of Michigan - DearbornDearborn, USA  
ckatkins@umd.umich.edu

Prithvi Thimmanayakanahalli Bachireddy  
College of Engineering and Computer Science  
University of Michigan - DearbornDearborn, USA  
prithvit@umd.umich.edu

*Abstract*—Every year, an average of 37 children die in the United States after being left in hot vehicles [1]. Each death is a tragedy that could have been averted with a simple and affordable engineering solution.

Keywords—automobile, vehicle, child, driver

Contents

[I. Introduction 1](#_Toc25354177)

[II. Related Work and State of the Industry 1](#_Toc25354178)

[III. Hardware Selection 1](#_Toc25354179)

[A. Microprocessor 1](#_Toc25354180)

[B. Sensors 1](#_Toc25354181)

[C. Other Hardware 2](#_Toc25354182)

[IV. Software Engineering 2](#_Toc25354183)

[A. Requirements 2](#_Toc25354184)

[B. Design Methodology 2](#_Toc25354185)

[C. Design 2](#_Toc25354186)

[V. Implementation Details 2](#_Toc25354187)

[A. Hardware Implementation 2](#_Toc25354188)

[B. Firmware Implementation 2](#_Toc25354189)

[C. Software Implementation 3](#_Toc25354190)

[VI. Functionality anD Testing 3](#_Toc25354191)

[A. Functional Requirements Testing 3](#_Toc25354192)

[B. Performance Requirements Testing 3](#_Toc25354193)

[C. Other Testing 3](#_Toc25354194)

[VII. User Manual and Maintenance 3](#_Toc25354195)

[VIII. Bill of Materials 3](#_Toc25354196)

[IX. Future Work 3](#_Toc25354197)

# Introduction

Modern vehicles are engineering wonders, containing more than 50 pounds of copper wire [2], increasingly sophisticated microprocessors, and safety systems that were unimaginable just ten years ago. The vehicles’ advanced sensors and controls allow vehicles to alert drivers to the presence of other vehicles, perform emergency braking for the driver, and even park and drive themselves. However, car manufacturers have been very slow to include systems that monitor the vehicle for children left alone. These systems are just beginning to be sold in the 2019 and 2020 model year vehicles [3] in many GM vehicles with several other manufacturers following suit. But what about parents that own older vehicles? Options exist, but many of them are video based, adding a device that could become a projectile in an accident. This paper describes a different alert system prototype that can be constructed using a simple microprocessor, simple sensors, and an internet connection that can monitor the vehicle for both a driver and a child that can be retrofitted to any vehicle.

# Related Work and State of the Industry

# Hardware Selection

## Microprocessor

During hardware selection several types of microprocessors were examined, including PIC processors, a Raspberry Pi, and Arduino. Ultimately an Arduino Nano 33 IoT was selected for several reasons. First, it is significantly easier to work with than the PIC, especially due to the fact that a WiFi connection was a requirement. The Arduino Nano IoT contains a WiFi and Bluetooth module on the board as well as included libraries, making this process much easier. The Arduino was also affordable, small, and had more than enough processing power to complete the required tasks.

## Sensors

Identify applicable funding agency here. If none, delete this text box.

The purpose of the sensors is to provide the Arduino with data about the presence of a driver and/or child in the vehicle. Strain gauges were selected to perform this task. A strain gauge is a piece of material (usually metal) that has a flexible backing attached [4]. When this material is bent even slightly, it alters the resistance of the device, producing a minor, but measurable, difference in the voltage across the device.

## Other Hardware

Identify applicable funding agency here. If none, delete this text box.

Due to the extremely small voltage change generated by applying force to a strain gauge, the Arduino requires some assistance. This comes in the form of a microchip called an HX711. This chip combines one or more strain gauges into a Wheatstone bridge [5] that amplifies the signal. In addition, the HX711 has a 24-bit ADC [6] for digitizing the signal with greater sensitivity than the Arduino’s 12-bit ADC.

# Software Engineering

## Requirements

1. System Requirements
   * [SYSFUNCT001] The system shall consist of hardware and software components that have the capability of monitoring a vehicle for the presence of a driver and child as well as the capability of alerting a user via text or email.
   * [SYSFUNCT002] The system shall have a user-configurable delay for signaling an emergency event.
   * [SYSPERF001] The system shall alert a user within 30 seconds (1-sigma) of an emergency event.
2. Embedded System Requirements
   * [ESFUNCT001] The hardware shall consist of an embedded system with one or more microprocessors, one or more sensors, and the hardware to support a wireless communication method (WI-FI or Bluetooth preferred).
   * [ESPERF001] The hardware shall transmit an alert to the backend system within 15 seconds (1-sigma) of an emergency event.
3. Backend Requirements
   * [BEFUNCT001] The software shall receive notifications wirelessly (using WI-FI or Bluetooth) from the hardware and transmit a notification to the user via text message or email.
   * [BEFUNCT002] The software shall be written in a high-level programming language such as C++, Java, or a similar language.
   * [BEPERF001] The software shall transmit an alert to the user within 15 seconds (1-sigma) of an alert received from the hardware.

## Design Methodology

The design method chosen for this project is the Agile system. The project was divided into three parts, each comprising several weeks. These parts are: requirements phase, design and implementation phase, and system integration and testing. The requirements phase was focused on identifying the system and subsystem requirements. This process involves identifying the needs of the system as well as the amount of available time, and creating requirements for the system that are both reasonable to accomplish and together create a functional system. The design and implementation phase involved the system-level design, including block diagrams and flow charts, as well as selecting the hardware that would be used for the project. The final phase involved combining the two parts of the systems and testing them for functionality and requirement validation.

[Describe Agile methods used here]

## Design

### Firmware Design

The firmware monitors the sensors for the presence of a child and, in the case of a child being left alone, sends an alert to the backend software that will then relay the alert to the user. Thus, the firmware must first monitor for the presence of a child. If no child is present, it is not required to operate at all and must periodically monitor if that status has changed. If a child is present, the system must monitor both the child and driver. The design also requires a timer before sending an alert to prevent false alarms if the driver is filling the car with fuel, loading or unloading the car, or getting out of the car to remove the child, which can take several minutes. The flow chart in Figure 1 shows the flow of the design.

No

Yes

Reset timer

Check Child

Check Driver

Yes

No

Send Alert

Timer

Yes

No

1. Check for child
2. Check for adult
3. If child present and driver is not, set a timer
4. Keep checking for driver or if child was removed
5. If child still alone at end of timer, send alarm

Figure : Firmware Flow Chart

### Software Design

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

# Implementation Details

The Vehicle Child Alert System consists of several distinct parts that have been divided into multiple parts. This includes the hardware design, consisting of the strain gauges, HX711 microprocessors, wires, and Arduino, the Firmware design, which consists of the code that runs on the Arduino to monitor the gauges and relay alerts to the backend, and the backend, which alerts the user in case of an emergency.

## Hardware Implementation

The hardware implementation connects the various pieces of the hardware physically. Figure 1 shows the wiring diagram for the system. In order for two 3-wire strain gauges to work properly, one white and one black wire from each gauges must be connected together and run to the HX711. The remaining white and black pair are similarly connected together. These wires are then attached to E+ or E- connection on the HX711 board. The red wires from the strain gauges are connected to the A+ and A- connections on the HX711 board, completing the circuit. The HX711 combines these connections to form the Wheatstone bridge internally. The HX711 is then connected to the Arduino with 4 wires. Power and Ground are connected to pins 2 and 14, respectively. The SCK (clock) and DT (Data Out) pins from the HX711 are then connected to two digital pins on the Arduino board. The Arduino board, in turn, is powered from a USB connection.

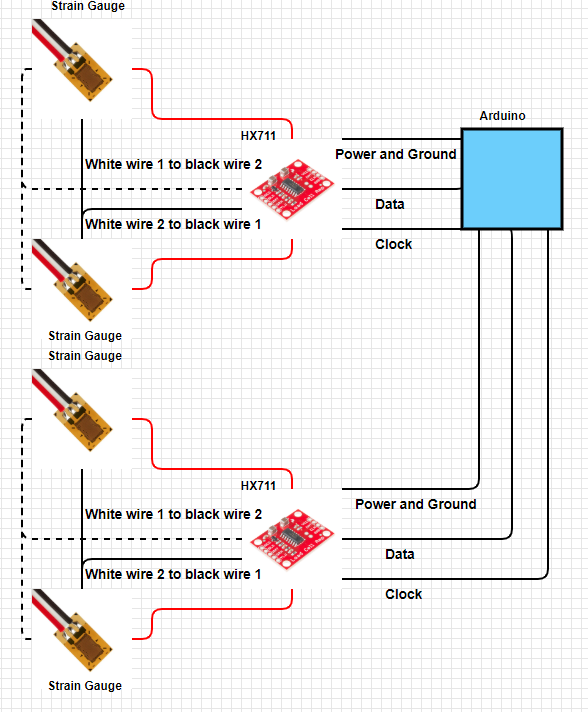


Figure : Hardware Wiring Diagram [drawn using 6]

## Firmware Implementation

#### Major Functions

The code is organized very simply, as most Arduino code is. There is a setup() function that performs all of the one-time operations. This includes opening the serial interface between the PC and the Arduino (when developing and debugging), setting up the HX711’s pin configurations, and connecting to the internet (in this case, WiFi).

The other major function is the loop() function. This is the typical continuous loop found in most embedded systems that contains most of the functional code. This section calls several other functions that monitor the sensors, check the WiFi status, and send an alert if necessary.

#### Minor Functions

The connectWifi() function performs the initial connection to the WiFi network, or can be called if a connection is lsot at any time. This uses the configured networks SSID and password to connect to a WPA2 encrypted WiFi network and blink the LED when it connects.

The dispWifiStatus() function displays the status of the WiFi connection to the serial monitor, including the network SSID and the Arduino’s assigned IP address.

The checkWifi() function checks that the Arduino is still connected to the network and tries to reestablish its connection if it has been lost using the connectWifi() function.

The checkDriver() and checkChild functions check the values from the sensors to determine if the driver or child is currently in the vehicle, returning a simple Boolean answer.

The checkSensors(int delayTime) function is the most important function of the code. This is the section where the logic that monitors the driver and child is located. See Figure 1 for the flow chart. This code consists of a while() loop that breaks when the logic determines that an alert should or should not be sent. It first checks for the presence of a child. If there is no child, it breaks the loop and returns to the loop() without sending an alert. If there is a child, it then checks for a driver. If a driver is present, then it breaks the while loop and returns to loop() without sending an alert. Otherwise, if there is a child and no driver, it delays for the specified delayTime before performing the checks again. If the driver has returned to their seat of the child has been removed from theirs, it breaks the while loop and returns to loop() with no alert being sent. Otherwise it returns a true for alert and an alert will be sent to the driver when it returns to loop().

## Software Implementation

The template is designed for, but not limited to, six authors**.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

# Functionality anD Testing

## Functional Requirements Testing

* [SYSFUNCT001] -
* [SYSFUNCT002] -
* [ESFUNCT001] -
* [BEFUNCT001] -

## Performance Requirements Testing

* [SYSPERF001] -

## Other Testing

An alert system can be built using simple sensors, a microprocessor, and an internet connection that monitors the vehicle for the presence of a driver and a child or children. When a child is left alone in the car for a period of time, an alert is sent to the parent or guardian

# User Manual and Maintenance

## User Setup

An alert system can be built using simple sensors, a microprocessor, and an internet connection that monitors the vehicle for the presence of a driver and a child or children. When a child is left alone in the car for a period of time, an alert is sent to the parent or guardian

## User Manual

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

## Maintenance

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

# Bill of Materials

|  |  |
| --- | --- |
| Item | Quantity |
| Arduino Nano IoT | 1 |
| 50 kg strain gauge | 4 |
| HX711 ADC | 2 |

# Future Work

## Temperature sensor

A temperature sensor connected to the Arduino would be a possible source for future work. This number could then be reported along with the alert message.

## Adjustable Alert Intervals

An adjustable alert interval is another future improvement that could be made. Ideally this would use the temperature sensor from the previous section to adjust the alert interval based upon the temperature inside the vehicle. This would provide an earlier alert when the temperature reaches more dangerous levels.

## Additional Sensors

Additional (preferably plug-and-play) sensors would improve the system, as they would allow for the monitoring of multiple children or even multiple adults. For instance, if an adult is in the passenger seat but the driver is not present (if the vehicle were getting refueled, for example), then there would be no need to send an alert. This would likely require code in the firmware to measure the amount of weight on the sensors and some threshold weight level that determines if the person is a child or an adult.

##### References

1. <https://www.cnn.com/2018/07/03/health/hot-car-deaths-child-charts-graphs-trnd/index.html>
2. <https://www.assemblymag.com/articles/92263-wire-harness-recycling>
3. <https://mashable.com/article/car-seat-alarms-prevent-hot-car-death>
4. <https://en.wikipedia.org/wiki/Strain_gauge>
5. <https://en.wikipedia.org/wiki/Wheatstone_bridge>
6. [<https://www.alldatasheet.com/datasheet-pdf/pdf/1132222/AVIA/HX711.html>](https://en.wikipedia.org/wiki/Wheatstone_bridge)
7. <https://www.digikey.com/schemeit/project/>