**Protecting Pet Safety using Temperature-Sensitive Collars**

ECE 4220 Final Project Proposal

April 1, 2015

Stephanie Cahail

**Problem**

When it’s either winter or summer, outdoor pets are vulnerable to extreme temperatures and without proper shelter and care can suffer from hyperthermia or hypothermia. This problem was chosen for which to find a solution because pets suffer from extreme weather conditions more often than they should due to owners just not being aware that their pets may be too cold or too hot. Although owners can tell if their pet is overheated or too cold based on how the animal is acting, it can be hard for owners to monitor the temperature of an animal because they’re not outside all the time or they’re not at their home. A solution for owners to be more aware of the condition of their pet(s) in extreme weather temperatures is to make a collar that can measure the temperature of the pet wearing it. If the temperature of the pet goes above or below what the normal temperature range should be, the owner would be notified via a central device to take action.

**Related Systems**

When considering a solution for this problem, it was pretty trivial that measuring the pet’s body temperature was necessary. How to go about obtaining temperature measurements, however, was not so trivial. The main inspiration for recording the temperature came from Biotrack’s method of sensing temperature differences in animals to determine hibernation patterns and signs of death. They track body temperature of animals by placing a thermistor under the animal’s tracking tag in a fashion so that the thermistor is touching the animal’s skin or fur [1]. Although this implementation is used to solve a problem that is different from the problem being addressed in this project, it is a method that could very easily be applied to the desired solution to the project’s problem.

Another factor needed for the solution is converting the output voltage of the thermistor to a temperature. An approach that was used to do so was built upon from an article on Adafruit. This article explained how to use a thermistor with an Arduino microcontroller. The method to calculate the thermistor temperature shown in this article was applied to the project, but altered to work with the TS-7250 board that will be used in the project rather than the Arduino board [2].

**Goals**

The solution will need to be performed on a smaller scale, so the first short-term goal for the project solution is to have a thermistor successfully measure temperatures. The second short-term goal is to be able to send temperature measurement readings to a computer through the TS-7250 board (the TS-7250 board is connected to the computer through a serial port). The final short-term goal is to successfully trigger alerts on the computer that contain a warning message and the pet’s current temperature when the temperature data sent to the computer is either higher or lower than the normal temperature range. Additionally, different types of warning messages shall be displayed based on the body temperature.

A long-term goal is to build on the project by implementing wireless communication between the microcontroller and a computer to trigger alerts on the computer. To accomplish this a different microcontroller would need to be used. Also, a smaller computer, such as a tablet, could be used. Additionally, another long-term goal is to design at least one physical collar that can successfully encompass the microcontroller and thermistor system without compromising the accuracy of the system. One last long-term goal is to be able to use multiple microcontrollers and thermistor circuits to simulate multiple collars being used simultaneously. Accomplishing these goals would allow the system to better resemble the collar and central device system solution desired.

The final goal is to design a collar that can measure the body temperature of the animal wearing it and wirelessly send the temperature data to a central device that will have a user interface to display any alerts along with the temperature reading of the pet if its temperature is above or below the normal temperature range for pets.

**Expected Outcome and Benefits**

The expected outcome is for the computer to display a warning message if the thermistor temperature reading is above or below the normal temperature range. The type of warning message displayed should depend on how far out of range the temperature is. The different types involve mild, moderate, and severe hypothermia and mild and severe hyperthermia. See the methodology section for more details. When the warning message is displayed, the computer should also display the current temperature reading.

The benefit of this expected outcome is that the owner is notified that their pet is in danger of getting hyperthermia or hypothermia, and to what degree that danger is, and he or she can then take special measures to keep pets safe from extreme temperatures.

**Specifications and Constraints**

Some constraints on the project are budget constraints and time constraints. In order to make this solution as cheap as possible, the only thing that will be purchased are the thermistors. Therefore, the only microcontroller that can be used is the one provided, which is the TS-7250 board. With that, the output will only be displayed on the computer that the TS-7250 board is connected to, so the wireless aspect desired will not be implemented in this project. Furthermore, due to only having one microcontroller, simulating multiple pets wearing temperature-sensitive collars will have to be done using multiple threads in the program instead of using different microcontrollers. Even if the budget constraint were not an issue, the little time available to successfully complete this project would not allow for a wireless implementation.

With that aside, the components that will be used for the project are a 10kOhm thermistor, a 10kOhm resistor, a TS-7250 microcontroller, and a computer. Figure 1 below shows the diagram for the system.

Computer

TS-7250 Microcontroller

R1:10kOhm

Thermistor

R2: 10kOhm Resistor

Figure 1: Flowchart for Project System

The idea of the system is to use a voltage divider to measure the voltage between the thermistor and the control resistor. The microcontroller will be used to both supply the input voltage (which can be any value less than six volts) and receive the output voltage value. The computer then uses that voltage value to find the thermistor’s resistance value, thus calculating the thermistor temperature. The computer, as mentioned earlier, will provide alerts based on whether or not the temperature value is out of range.

**Assumptions**

There are some assumptions that need to be made about the project. First of all, the microcontroller and circuit are assumed to be a part of the pet collar, while the computer is assumed to be the central device on which the owner would receive alerts. Additionally, the room temperature is assumed to be 72 degrees Fahrenheit, and the beta coefficient of the thermistor is assumed to be 3500. The room temperature and the beta coefficient of the thermistor will be used in further equations, which will be discussed in the next section. Also, the materials used to obtain various thermistor temperature readings are assumed to be temperatures of animals in extreme conditions. Lastly, the use of multithreading is assumed to represent the use of multiple collars simultaneously.

**Methodology**

The computer uses the thermistor voltage value to find the thermistor’s resistance value. The equation used for that is:

Vout = (R1/(R1 + R2)) \* Vin

Where Vin is the input voltage, R1 is the thermistor resistance that needs to be calculated, R2 is the control resistance value, which is 10kOhms in this case, and Vout is the output voltage.

Once the thermistor’s resistance value is found, the temperature can then be calculated as so:

1/T = (1/To) + (1/B) \* ln(R1/R2)

Where T is the calculated temperature, To is the room temperature, and B is the beta coefficient of the thermistor [2].

After calculating the thermistor temperature it must then be determined whether or not the temperature is within the normal range. The normal temperature range for a dog or a cat is 100.5 to 102.5 degrees Fahrenheit [3]. Therefore, when the temperature measurement is not within that range a warning should be displayed with the temperature reading. If the temperature is less than 100.5 degrees but greater than 90 degrees Fahrenheit, a mild hypothermia warning message should be displayed. If the temperature is less than 90 degrees but greater than 82 degrees Fahrenheit then a moderate hypothermia warning message should be displayed. If the temperature is less than 82 degrees Fahrenheit then a severe hypothermia warning message should be displayed [4]. In term of hyperthermia, if the temperature is greater than 102.5 degrees but less than 105 degrees Fahrenheit then a mild hyperthermia warning message should be displayed. If the temperature is greater than 105 degrees then a severe heat stroke message should be displayed [5].

When collecting temperature readings, the thermistor will be placed next to mediums of various temperatures in order to collect a variety of data.

Some concepts from ECE 4220 that will be applied in this project are multithreading and thread synchronization, inter-process communication, and the first-come-first-serve concept of the time-shared real time scheduling algorithm. One of the ideals of the project is to simulate multiple outdoor pets wearing the collars. This resembles a pet owner having more than one outdoor pet. With that, task scheduling needs to be considered. Ideally, the first pet whose temperature measurement is out of range is the pet that the owner needs to be alerted about, so it makes sense that the first-come-first-serve idea of the time-shared real time scheduling algorithm. In order to have direct access to the TS-7250 board the data-collecting functionality should be performed in a kernel module. Since the thermistor temperature calculations and alert handling will be in the user space, communication between the kernel module and the user space is essential, which covers the concept of inter-process communication. In order to simulate someone owning multiple outdoor pets with the collar implementation, multiple threads are used to represent multiple collars. When the output voltage is sent to the user space from the kernel through a pipe, a thread is spawned to perform the calculations for the thermistor temperature associated with the voltage read in. In order to control the number of threads spawned a predetermined and customizable cap value and an incremented counter are used. With that, the concept of multithreading is covered. The challenge of multithreading is ensuring that the expected outcome is not affected by it. In this case, to ensure that the alerts caused by different threads are being printed accurately, mutual exclusion will be used to make sure an alert from one thread is completely done being printed before an alert from another thread starts to print. This method embraces the concept of thread synchronization.

FairCom will not be used in this project.

**Timeline and Milestones**

By April 16, 2015 the first draft of the coding for the project should be completed in order to begin testing.

By April 30, 2015 the project should be functional without the thread implementation. (i.e. simulating one collar in the implementation should work).

By May 7, 2015 the project should be fully functional with the thread implementation in order to start writing the report. \*

By May 15, 2015 the final project report should be completed as well as the demonstration.

\*The time frame from May 7 to May 15 will allow a little room for last minute debugging in the code if needed.

**Strengths and Limitations**

The strength of the system is that it provides the basic functionality of reading the thermistor temperature and displaying warnings based on the temperature. The limitation is that the system is not wireless, so it cannot be used as is to measure animal body heat and alert owners. However, since the basics are already implemented, it will not be that difficult to build upon in the future to accomplish the long-term and final goals discussed earlier.

**Works-Cited**

[1] Biotrack, 'Temperature Sensing'. [Online]. Available: http://www.biotrack.co.uk/pdf/sensetemp.pdf.

[2] Learn.adafruit.com, 'Using a Thermistor | Thermistor | Adafruit Learning System'. [Online]. Available: https://learn.adafruit.com/thermistor/using-a-thermistor.

[3] Pet Health Network, 'Hypothermia and Your Pet', 2012. [Online]. Available: http://www.pethealthnetwork.com/dog-health/dog-diseases-conditions-a-z/hypothermia-and-your-pet.

[4] Canidae Blog, 'Hypothermia and Your Pet: What are the Signs?', 2011. [Online]. Available: http://www.canidae.com/blog/2011/12/hypothermia-and-your-pet-what-are-signs.html.

[5] Petmd.com, 'Cat Heatstroke Causes - Heatstroke Symptoms in Cats | petMD'. [Online]. Available: http://www.petmd.com/cat/emergency/common-emergencies/e\_ct\_heat\_stroke. [Accessed: 01- Apr- 2015].