Livestock Temperature Monitor

By

Cain Benink
Michael Goldstein
Yue Wang

Design Document for ECE 445, Senior Design, Spring 2017

TA: Daniel Gardner

24 February 2017

Project No. 4
Contents

1 Introduction 4
   1.1 Objective and Background 4
   1.2 High Level Requirements 4

2 Design 5
   2.1 Block Diagram 5
   2.2 Block Descriptions 5
      2.2.1 Sensor Block 5
         2.2.1.1 Thermistor 6
         2.2.2.2 Resistor 7
      2.2.2 Control Block 7
         2.2.2.1 Low Power RTC 8
      2.2.3 Transmitter Block 8
         2.2.3.1 Analog to Digital Converter 8
         2.3.3.2 RF Transmitter 8
         2.3.3.3 Antenna 8
      2.2.4 Power Block 8
         2.2.4.1 Battery 8
      2.2.5 Receiver Unit 8
         2.2.5.1 Surge Protector 9
         2.2.5.2 Raspberry Pi 9
         2.2.5.3 RF Receiver 9
         2.2.5.4 Antenna 9
      2.2.6 Tolerance Analysis 9
   2.3 Risk Analysis 9
3. Design Requirements and Verification

3.1 Sensor Block

3.1.1 Thermistor

4 Costs

4.1 Parts

4.2 Labor

4.2.1 Research and Development

4.2.2 Assembly

4.3 Grand Total

5 Schedule

6 Ethical Considerations


6.2 Safety Guidelines

6.3 IACUC POLICY [12]

References

Appendix A Requirement and Verification Table

Appendix B Circuit Schematic and PCB Drawing
1 Introduction

1.1 Objective and Background
Current methods of diagnosing sick animals are insufficient. The process is largely manual. We plan to change that by offering a solution to wirelessly monitor the core temperature of cattle.

According to the Wall Street Journal, “To monitor cattle health, feedlots typically rely on cowboys to ride through pens, watching for lethargic animals that are having trouble breathing.”[1] Because the process is labor intensive and the symptoms identifiable show up days after the onset of disease, the livestock industry suffers from a great deal of both false positive and false negative sickness. This results in a great deal of animal death, over medication of animals, unnecessary vet visits, and unnecessary loss of organic certification. This problem costs the cattle industry over 2 billion dollars in losses annually. As of 2016, there are about 92.0 million heads of cattle in the United States.[2] Of these cows, digestive and respiratory illness, mastitis, and other diseases kill approximately 1.5 million of them.[3]

We aim to fix these problems by creating a cow tag that will constantly measure the temperature of the animal and send it back to a receiver via RF 915MHz. We chose this band because it is unregulated in the United States and offers the best transmission range and bandwidth of the unregulated RF bands. The receiver would upload data via the internet and a GUI would alert the farmer of animals that are running fevers. This solution leads to less labor intensive, more accurate, and earlier diagnosis of sick animals. Because of the amount of money farmers lose to this problem, a working solution would be easily marketable -- buying our product would literally save them money.

1.2 High Level Requirements

● Units must be durable enough and have a battery life long enough for the unit to last the entire lifespan of a beef cow (about 18 months).

● Units must transmit temperatures frequently enough and with high enough resolution to detect a cow’s fever within 12 hours.

● Units must be as low cost to manufacture as possible, ideally less than $10.
2 Design

2.1 Block Diagram

2.1.1 Tag Block Diagram

Refer to Appendix B for a schematic of the design.
2.1.2 Receiver Block Diagram

![Block Diagram of Receiver](image1)

2.1.3 Physical Specifications

![Physical CAD Drawing of Cow Tag](image2)

Figure 3 diagrams the physical dimensions of a cow tag thermometer. The tag will be a total of 5 inches tall, 3 inches wide, and 10mm deep. The thermistor will protrude out of the plastic housing 6 inches in order to reach the ear canal of the cow. The 10 mm deep is to have enough room for the ~2.5mm of board and be thick enough to protect the electronics. The hole at the top of the tag is for
the piercing that the cow receives. The piercer keeps the tag connected to the cow’s ear.

2.2 Block Descriptions

2.2.1 Sensor Block
The sensor block is a thermistor and a regular resistor in series with each other. By reading the voltage between them, we can use that value to calculate the temperature the thermistor is currently sensing. The power input from the control block to the block is Vcc, which in this design is 3V, and the block will output a voltage that will depend on the temperature of the thermistor.

![Sensor Block Schematic](image)

Figure 2.4: Sensor Block Schematic

2.2.1.1 Thermistor
The circuit uses a NXFT15WB473FA2B150 thermistor made by Murata [4], with important specifications in the table below:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 C Resistance</td>
<td>47 kΩ</td>
</tr>
<tr>
<td>25 C B Value</td>
<td>4050 K</td>
</tr>
<tr>
<td>Tolerance</td>
<td>1%</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-40 C to +125 C</td>
</tr>
</tbody>
</table>

Table 2.1: Important Specifications of the Thermistor [4]

The formula to calculate the resistance from temperature is as follows [5]:

\[
R = R_0 e^{-\frac{B}{T_o} e^{\frac{B}{T}}}
\]  

(1)

Knowing that the average temperature of a cow is 38.6 C and goes up to 40 C in a fever [6], we can
calculate the resistance range of the thermistor by putting those values into equation 1.

\[ R_{\text{min}} = 47000\Omega \times e^{-4050K/298.15K}e^{4050K/313K} = 24672\Omega \]  
\[ R_{\text{max}} = 47000\Omega \times e^{-4050K/298.15K}e^{4050K/311.6K} = 26149\Omega \]  

That means the resistor in series with the thermistor should ideally be the average of the two calculated values above, so 25409. Unfortunately they do not make resistors that accurate so the design will settle with a 25.5kΩ resistor.

We can then use the voltage divider rule along with the formula for temperature to generate a graph for the Temperature based on the voltage output from the voltage divider.

\[ V_{\text{out}} = \frac{R}{R+25500} \]  
\[ T = \frac{B}{\ln(R/R_o)-B/T_o} \]

![Voltage Divider Output vs Temperature](image)

**Figure 2.5: Voltage Output vs Thermistor Temperature**

The graph generate above shows us that for the range of temperature we are interested, 38.6 C to 40 C, the voltage output from the voltage divider will lie between 1.45 V and 1.55 V.

### 2.2.2.2 Resistor

The resistor component is there simply so that we can measure voltage to eventually calculate the temperature of the thermistor. Its value has already been calculated above and actual value should be 25.5 kΩ.
2.2.2 Control Block
The control logic keeps track of time so that it knows when to power up the other components, and also controls how long the power and data is being sent to the other components. We also will have a 4 byte identifier value stored in the microchip upon programming. That value along with a 4 byte temperature and a 2 byte CRC will be transmitted to the receiver (so the entire packet is a total of 10 bytes). The control block itself is built into an CC1310 chip along with other components for optimal power efficiency. The control block gets 3V of power from the battery.

2.2.2.1 Low Power RTC
The RTC is built into the CC1310 chip and provides accurate time tracking capabilities while keeping power consumption to a minimum. The RTC will be kept to a 32.8 kHz clock cycle which means that it needs to count to $32800 \times 60 \text{ seconds/minute} \times 5 = 9840000$ in order to keep track of 5 minute intervals.

2.2.2.2 Analog to Digital Converter
The analog to digital converter is also built into the CC1310 and receives the voltage output from the sensor module and reads it into the microcontroller. The CC1310 ADC is 12 bits, and taking into consideration the Vcc of 3V, leaves an error range of $\pm 0.000366 \text{V}$.

2.2.3 Transmitter Block
The transmitter block is in charge of preparing the measurement for transmission, and eventually transmitting the measurement through the antenna. In the CC1310 chip, all the necessary components for transmission is already built in except for the antenna.

2.3.3.1 RF Transmitter
The RF transmitter is built into the CC1310 chip. With the variety of available frequencies that the CC1310 chip is able to transmit at, we picked 915 MHz because it was in the public range. The RF transmitter will take the voltage signals from the microcontroller and combines it with the 10 bytes of identification data to produce a 13 byte package for final transmission.

2.3.3.2 Antenna
Since we picked the transmission to be 915 MHz, the antenna will be a 915 MHz capable antenna. The cheapest ones at 915 MHz provide a -4dB gain output which is enough for the 100m range desired [7].

2.2.4 Power Block
The power unit provides the necessary power to run all the components of the ear tag transmitter. It is essentially a battery that always powers the control unit but the control unit controls the power supply to the rest of the ear tag transmitter components.

2.2.4.1 Battery
This is the battery that provides power for the ear tag transmitter. It provides power to the control unit as well as the clock, and the control unit is in charge of sending power to all the other
components of the transmitter.

The calculations for how much power is needed to power the transmitter unit for two years is as follows:

Microcontroller Specifications[8]:

--Wide Supply Voltage Range: 1.8 to 3.8 V

– RX: 5.4 mA

– TX at +10 dBm 868 MHz: 13.4 mA

– Active-Mode MCU 48 MHz Running Coremark: 2.5 mA (51 µA/MHz)

– Active-Mode MCU: 48.5 CoreMark/mA

– Active-Mode Sensor Controller at 24 MHz: 0.4 mA + 8.2 µA/MHz

– Sensor Controller, One Wakeup Every Second Performing One 12-Bit ADC Sampling: 0.95 µA

– Standby: 0.7 µA (RTC Running and RAM and CPU Retention)

– Shutdown: 185 nA (Wakeup on External Events)

Message is 13 bytes.

Send message every (s) seconds (s) = sec/msg

Standby in between messages

Wake up time is 174 us

Conservatively we are assuming a 1 ms wake up and set up time.

Standby power:

Conservatively it's constantly using standby power.

(.7 µA)(2 years)(365 days)(24 hours) = .0123 Ah (6)

Message Transfer Power Usage:
\[
\text{((12 bytes)/(50 kb/s))} = .00192 \text{ seconds/message} \\
((.00192 \text{ sec/msg})(13.4 \text{mA})) = 25.73 \text{ uA/Msg} \\
(1/(s) \text{ msg/sec})(25.73 \text{ uA/Msg}) = 25.73/(s) \text{ uA} \\
(25.73/(s) \text{ uA/s})(2 \text{ years})(365 \text{ days})(24 \text{ hours}) = .451/(s) \text{ Ah} \quad (7)
\]

Wake up/Set up Time Power Consumption:

Duty Cycle = (1 ms/msg)(1/(s) msg/s) = .001/(s) \\
((.001/(s))(2.5 mA)) = 2.5/(s) \text{ uA} \\
(2.5/(s) \text{ uA/s})(2 \text{ years})(365 \text{ days})(24 \text{ hours}) = .0438/(s)\text{Ah} \quad (8)

Active Power Usage:

(1/(s) msg/s)(.00192 s/msg) = .00192/(s) \text{ duty cycle} \\
(.00192/(s) \text{ duty cycle})(2.5 mA) = 4.8/(s) \text{ uA} \\
(4.8/(s) \text{ uA})(2 \text{ years})(365 \text{ days})(24 \text{ hours}) = .0841/(s) \text{ Ah} \quad (9)

Thermistor Power Usage During Transfer:

\begin{align*}
\text{ohms} &= 25.5\text{kohms}(\text{Thermistor}) + 25.5\text{kohms}(\text{Series Resistance}) \\
\text{Voltage} &= 3.3\text{V} \\
(3.3\text{V})/(51\text{kohms}) &= 64.7\text{uA} \\
\text{Duty Cycle} &= .00192/(s) \\
(.00192 \text{ duty cycle})(64.7 \text{ uA}) &= 124.2 \text{ nA} \\
(124.2 \text{ nA/s})(2 \text{ years})(365 \text{ days})(24 \text{ hours}) &= .002176/(s) \text{ Ah} \quad (10)
\end{align*}
Total Amp-hours:

\[(\text{Standby}) + (\text{Thermistor}) + (\text{Transfer}) + (\text{Setup}) + (\text{Active}) = \text{Total Ah} \]

\[.0123 \text{ Ah} + .002176/(s) \text{ Ah} + .451/(s) \text{ Ah} + .0438/(s) \text{ Ah} + .0841/(s) \text{ Ah} \]

\[= .0123 + .5811/(s) \text{ Ah} \]

(11)

Since we are planning on having the device transmit every 5 minutes, then \((s)\) would be 300, so total power consumption in two years using equation 11 is:

\[.0123 + .5811/300 \text{ Ah} = 0.01424 \text{ Ah} \]

So the transmitter needs a battery that outputs 3V and has at least 14.24 mAh, and button batteries satisfy that condition. We picked the Panasonic CR2025 which has output of 3V and capacity of 165 mAh, it can be purchased for as low as 30 cents and widely available everywhere[9]. That particular button battery will allow the transmitter to theoretically run up to 20 years.

### 2.2.5 Receiver Unit

The receiver unit is in charge of collecting data from all the various ear tag transmitters and process them to be present to the customer. It includes an antenna attached to a RF Receiver chip to receive signals, a Raspberry Pi 3 to process and store the data, and a surge protector connected to a conventional outlet for power.

#### 2.2.5.1 Raspberry Pi 3

This is the calculating component of the receiver unit. It takes the measurement reading signals from the RF receiver and calculates the temperature that would cause those voltages. It can then store the data in its memory and later pass the data on through the built in USB adapter. Using the processed data, we can run a webserver on the Raspberry Pi 3 and also use the Wifi capability to transmit data to a bigger central database if needed. Since the receiver unit is plugged into a wall, this is always on and ready to receive new data.

#### 2.2.5.2 RF Receiver

The RF receiver is always on and ready to receive new data from any of the ear transmitter’s RF transmitters. The RF Receiver receives data from the antenna and any data it receives will be first processed by the RF Receiver and then sent to the raspberry pi unit for final conversion.

#### 2.2.5.3 Antenna

The antenna takes the received signal and passes it to the RF Receiver chip. The transmitter antenna is already set at 915 MHz, so we will use the same one on the receiving end.
2.2.6 Tolerance Analysis

For the tolerance analysis, we will analyze the margin of errors for the sensor block, which is the most crucial part of the whole design. If the initial measurement of the temperature is off, none of the other components matter since it’ll all be wrong in the end.

Both the resistor and thermistor comes with 1% tolerance in resistance, and thermistor also has a 1% tolerance on the B value.

The ADC of the microcontroller is 12 bits and maps a voltage range of 0 to Vcc (3V). This means the distance between any two digital values is 3/2^12 volts. The maximum aliasing error occurs when an analog value is exactly halfway between two digital values. Thus, the aliasing error is 3/2^13 volts, which evaluates to 0.000366 volts.

Assuming that the ADC aliasing error is at max, and tolerance error is also at max, we can calculate the temperature difference from the theoretical.

From the sensor block we have the formulas:

\[ V_{out} = \frac{R}{R + 25500} \]  \hspace{1cm} (12)

\[ T = \frac{B}{\ln(R/R_0) - B/T_0} \]  \hspace{1cm} (13)

We can then vary the Vout and 25500 ohm resistor value in equation 12 along with the B value in equation 13 to find tolerances. Using python, we have the code for plotting below:

```python
import matplotlib.pyplot as plt

R = lambda v: 25500*v**1.0/(3-v)
T = lambda r: (4050.0/np.log(r**1.0/(47000*np.exp(-4050/298.15)))-273)

Reu = lambda v: 25500*0.99*(v-0.000366)**1.0/(3-(v-0.000366))
Teu = lambda r: (4050.0*0.99/np.log(r**1.0/(47000*0.99*np.exp(-4050*0.99/298.15)))-273)

Rel = lambda v: 25500*1.01*(v+0.000366)**1.0/(3-(v+0.000366))
Tel = lambda r: (4050.0*1.01/np.log(r**1.0/(47000*1.01*np.exp(-4050*1.01/298.15)))-273)
```

**Figure 2.6: Tolerance Formulas for Tolerance Calculations**

In the code, R and T are the theoretical perfect values with no error in tolerance. Reu and Teu will give the maximum amount of Temperature that can result from all the errors combined, and Rel and Tel will give the minimum amount of Temperature that can result from all the errors combined. This code is then used to generate the following graph:
We can see in the above figure that in the 38.6 C to 40 C range, the error range is pretty small.

We can also use the code in figure 2.6 to see the error range at specific voltage outputs.

At this voltage: 1.4 Theoretical: 42.4508769949 Upper: 42.6480628029 Lower: 42.2578351942
At this voltage: 1.42 Theoretical: 41.7946627699 Upper: 41.9843865314 Lower: 41.6089179007
At this voltage: 1.44 Theoretical: 41.142102621 Upper: 41.3244396777 Lower: 40.9635816151
At this voltage: 1.46 Theoretical: 40.4929349243 Upper: 40.6679570978 Lower: 40.3215681419
At this voltage: 1.48 Theoretical: 39.8469036728 Upper: 40.0146793865 Lower: 39.6826247832
At this voltage: 1.5 Theoretical: 39.2037578782 Upper: 39.3643522719 Lower: 39.0465037495
At this voltage: 1.52 Theoretical: 38.5632509968 Upper: 38.716726032 Lower: 38.412961594
At this voltage: 1.54 Theoretical: 37.925140374 Upper: 38.0715549317 Lower: 37.781758664
At this voltage: 1.56 Theoretical: 37.2891867056 Upper: 37.4285966764 Lower: 37.1526585702
At this voltage: 1.58 Theoretical: 36.655153512 Upper: 36.7876118795 Lower: 36.5254276675
At this voltage: 1.6 Theoretical: 36.0228066238 Upper: 36.1483635408 Lower: 35.899834547

The above figure shows us that within the range we want, 38.6 C to 40 C, the error range is at maximum +/- 0.15C around the theoretical value, which is within acceptable limits.

2.3 Risk Analysis

We believe that the transmission/receiving module poses the greatest risk potential in our Livestock Temperature Monitor. It has the greatest risk potential because it is the most likely to fail and if it fails it will have a big impact on its function. The transmission module is the most complex portion of our design, and, therefore, it has more components, systems, etc. that can fail than any other module.
Also, when the transmission module fails it will not transmit the data correctly which jeopardizes the intended function.

To mitigate some of the risks of the transmission module, we will design the system to send multiple messages of the current value every time it is transmitting the data. Another mitigation is to have parity bits within the messages that we can check against for errors.

The tolerances of the transmission module are related to our first and second high level requirements. As for durability and battery, the module should be able to survive any temperature that a cow is expected to survive outdoors, 32-90 degrees Fahrenheit. The module should also be able to survive any forces that a normal livestock tag can survive currently. The battery must not be drained by the module before the required time. As for requirement two, the tag must be within 0.1 degrees Celsius of the actual temperature, transmit up to 100m and report to the receiver once an hour. Following these tolerances can mitigate the risk of a failing transmission/receiving module.
3. Design Requirements and Verification

3.1 Sensor Module

Table 2: Sensor Module Requirements and Verification

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
<th>Verification status (Y or N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The Sensor shall measure temperature by outputting a voltage that is related to the sensed temperature.</td>
<td>1. The sensor shall be able to accurately measure temperature.</td>
<td></td>
</tr>
<tr>
<td>a. The Sensor shall output a voltage range from 1.45V to 1.55V, +/- .01V, linearly for the temperature range 38.6 C to 40 C respectively.</td>
<td>a. The thermistor will be put in a temperature chamber. The chamber temperature will be varied from 38.6 C to 40 C over 4 hours, starting at 38.6 and increasing at a rate of .35 C/hour. We will use a multimeter to measure the output voltage. If the sensor outputs 1.45V to 1.55V, +/- .01V, linearly over the temperature range it passes.</td>
<td></td>
</tr>
<tr>
<td>2. The Sensor shall be able to reach into a cow’s ear.</td>
<td>2. The Sensor shall be able to reach into a cow’s ear.</td>
<td></td>
</tr>
<tr>
<td>a. The sensor shall be able to reach 6 inches, +/- .2 inches, from the ear tag.</td>
<td>a. We will use a tape measure to measure from the base of the sensor location on the board to the tip of the sensor. If the length is 6 inches, +/- .2 inches, it passes.</td>
<td></td>
</tr>
</tbody>
</table>

3.2 Power Module

Table 3: Power Module Requirements and Verification

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
<th>Verification status (Y or N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The power module shall provide power to the device for 2 years, +/- .1 years, in temperatures that cows can survive outside, as low as 0 C[10].</td>
<td>1. The power module shall power the device for no less than 2 years.</td>
<td></td>
</tr>
<tr>
<td>a. The power module shall</td>
<td>a. We will run the device for 4 hours and measure the total Amp-hours used with a wattmeter. If the device</td>
<td></td>
</tr>
</tbody>
</table>
provide 20mAh total, +/- 1mAh, over 2 years, +/- .1 years.

b. The power module shall be able to provide 13.4mA, +/- .1mA, in temperatures 0 C to 50C.

c. The power module shall provide a constant 3V, +/- .1V.

uses less than 4.57 uAh, +/- .2 uAh, then the module passes.

b. The device, active and demanding 13.4mA, will be put in a temperature chamber that will vary the temperature from 0C to 50C over 4 hours, rate of 12.5 C/hour, and power measurement, with a wattmeter, will be taken. If the power module outputs 13.4mA it passes.

c. We will run the device for 4 hours and measure output voltage with a multimeter. If the power module supplies a constant 3V, +/- .1V, then the module passes.

3.3 Control Module

**Table 4: Control Module Requirements and Verification**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
<th>Verification status (Y or N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The control module software shall receive sensor data.</td>
<td>1. The control module shall receive sensor data.</td>
<td></td>
</tr>
<tr>
<td>a. The control module software shall activate every 5 minutes, +/- 1 second.</td>
<td>a. Run the device for 30 minutes and using an oscilloscope, track the input voltage and current. If we observe a change in current from the standby current, .7µA, to active/transmit power, 13.4mA, every 5 minutes, +/- 1 second, then it passes.</td>
<td></td>
</tr>
<tr>
<td>b. The control module software shall allow a 1ms, +/- .01ms, wake up time before each processing of sensor data.</td>
<td>b. Run the device for 30 minutes and using an oscilloscope, track the input voltage and current. If we observe a 1ms, +/- .01ms, active current draw,</td>
<td></td>
</tr>
<tr>
<td>c. The control module software shall go into standby mode after transmission is finished.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. The control module shall package the sensor data.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a. The control module</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
software shall convert the analog thermistor voltage data into 4 bytes of digital data and append the voltage data onto the 4 byte ID data to make an 8 byte packet.

3. The controller module shall hold in memory a unique ID.
   a. The controller module shall use memory to save a unique 4 byte ID.

2.5mA, before a transmitting current draw, 13.4mA, then it passes.

c. Run the device for 30 minutes and using a oscilloscope track the current draw. If between the active/transmit cycle, every 5 minutes +/- 1 second, the current draw is the standby current, 0.7 µA, then it passes.

2. The control module shall package the sensor data.
   a. Run the device for 30 minutes and use an oscilloscope to track the “tx” output pin. If every 5 minutes, +/- 1 second, the “tx” output pin outputs an 8 byte packet it passes.

3. The controller shall hold in memory a unique ID.
   a. Run the device for 30 minutes and use an oscilloscope to track the “tx” output pin. If every 5 minutes, +/- 1 second, the “tx” outputs and the first 4 bytes are the unique ID it passes.

### 3.4 Transmitter Module

**Table 5: Transmitter Module Requirements and Verification**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
<th>Verification status (Y or N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The transmitter module software shall transmit the packaged data.</td>
<td>1. The transmitter module software shall transmit the packaged data.</td>
<td>Y</td>
</tr>
<tr>
<td>a. The transmitter module software shall transmit the 8 byte packet at</td>
<td>a. Run the device for 30 minutes and use an MXA signal analyzer to look for a</td>
<td>N</td>
</tr>
<tr>
<td>50kb/s at the 915Mhz ISM band at 10dBm, +/- .1dBm.</td>
<td>signal on the 915MHz band. If every 5 minutes, +/- 1 second, an 8 byte</td>
<td></td>
</tr>
<tr>
<td>b. The antenna shall be able</td>
<td>packet is detected it</td>
<td></td>
</tr>
</tbody>
</table>
to transmit data 100m, +/- 5m, on the 915MHz ISM band at 10dBm, +/- .1dBm.

b. Run the device for 30 minutes 100 meters away from the MXA signal analyzer that is looking for a signal on the 915MHz band. If every 5 minutes, +/- 1 second, an 8 byte packet is detected it passes.

3.5 Receiver Module

Table 5: Receiver Module Requirements and Verification

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
<th>Verification status (Y or N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The receiver module shall receive the packaged data.</td>
<td>1. The receiver module shall receive the packaged data.</td>
<td></td>
</tr>
<tr>
<td>a. The antenna shall be able to receive data 100m, +/- 5m, on the 915MHz ISM band.</td>
<td>a. Run a RF signal generator and push a signal to 915MHz 100m away from the receiver. If the receiver antenna, measured by an oscilloscope, receives the transmitted data it passes.</td>
<td></td>
</tr>
<tr>
<td>2. The receiver module software shall process the package into temperature and ID.</td>
<td>2. The receiver module shall process the package into temperature values.</td>
<td></td>
</tr>
<tr>
<td>a. The raspberry pi software shall process the 8 byte data into the ID data and Temperature data.</td>
<td>a. Send an 8 byte signal using a RF signal generator over the 915MHz band, and if the raspberry pi separates the signal into two 4 byte packages, ID and temperature, it passes.</td>
<td></td>
</tr>
<tr>
<td>3. The receiver module software shall convert the 4 byte temperature data into a temperature value.</td>
<td>3. The receiver module software shall convert the 4 byte temperature data into a temperature value.</td>
<td></td>
</tr>
<tr>
<td>a. The raspberry pi software shall process the 4 byte temperature data into a temperature value.</td>
<td>a. Input a 4 byte value into the raspberry pi and if the software converts the bytes into a temperature value it passes.</td>
<td></td>
</tr>
</tbody>
</table>
## 4 Costs

### 4.1 Parts

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Part Type</th>
<th>Unit Px 100</th>
<th>Unit Px 5000</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texas Instruments</td>
<td>CC1310F32RHBR</td>
<td>Microcontroller</td>
<td>$3.740</td>
<td>$2.200</td>
<td>TI Store</td>
</tr>
<tr>
<td>Panasonic</td>
<td>CR2025</td>
<td>Battery</td>
<td>$0.222</td>
<td>$0.159</td>
<td>Mouser</td>
</tr>
<tr>
<td>Linx</td>
<td>BAT-HLD-001</td>
<td>Battery Holder</td>
<td>$0.233</td>
<td>$0.183</td>
<td>Mouser</td>
</tr>
<tr>
<td>Murata</td>
<td>NXFT15WB473FA28150</td>
<td>Thermistor</td>
<td>$0.274</td>
<td>$0.172</td>
<td>Mouser</td>
</tr>
<tr>
<td>Panasonic</td>
<td>ERJ-2RKF2552X</td>
<td>Resistor</td>
<td>$0.006</td>
<td>$0.003</td>
<td>Mouser</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$4.475</strong></td>
<td><strong>$2.717</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Quote Description</th>
<th>Unit Px 100</th>
<th>Unit Px 5000</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB</td>
<td>50x40mm, 1.6mm thick, 6mil spacing</td>
<td>$4.900</td>
<td>$0.931</td>
<td>PCBWay</td>
</tr>
<tr>
<td>Housing</td>
<td>Plastic Case, 3D printed ABS</td>
<td>$3.938*</td>
<td>$3.938*</td>
<td>Illinois MakerLab</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>$8.838</strong></td>
<td><strong>$4.869</strong></td>
<td></td>
</tr>
</tbody>
</table>

*Estimate based on $0.15/g * 1.05g/cc * 25cc until we have a CAD drawing and get a quote*

<table>
<thead>
<tr>
<th>Unit Price, 100</th>
<th>$13.313</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Price, 5000</td>
<td>$7.586</td>
</tr>
<tr>
<td>Lot Price, 100</td>
<td>$1,331.250</td>
</tr>
<tr>
<td>Lot Price, 5000</td>
<td>$37,928.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Part Type</th>
<th>Unit Px 1</th>
<th>Unit Px 100</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi</td>
<td>Pi 3 Model B</td>
<td>Computer</td>
<td>$39.950</td>
<td>$39.950</td>
<td>Adafruit</td>
</tr>
<tr>
<td>Johanson</td>
<td>0915AT43A0026E</td>
<td>Antenna</td>
<td>$1.280</td>
<td>$0.574</td>
<td>Mouser</td>
</tr>
<tr>
<td>Kingston</td>
<td>SDC10/4GB</td>
<td>Micro SD Card</td>
<td>$6.490</td>
<td>$6.490</td>
<td>Amazon</td>
</tr>
<tr>
<td>CanaKit</td>
<td>Pi MicroUSB Supply</td>
<td>Power Supply</td>
<td>$7.990</td>
<td>$7.990</td>
<td>Amazon</td>
</tr>
<tr>
<td>Sparkfun</td>
<td>COM-13909</td>
<td>RF Transmitter</td>
<td>$4.950</td>
<td>$4.950</td>
<td>Mouser</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$60.66</strong></td>
<td><strong>$59.954</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Table 4.5  Unit & Lot Costs - Receiver Unit

<table>
<thead>
<tr>
<th>Unit Price</th>
<th>Lot Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>$60.660</td>
<td>$5,995.400</td>
</tr>
<tr>
<td>$59.954</td>
<td></td>
</tr>
</tbody>
</table>

### 4.2 Labor

#### 4.2.1 Research and Development

We have three team members doing research and development for approximately 20 hours per week for about 14 weeks. $2.5 \times 3 \times 20 \times 14 \times $45 yields a research and development cost of approximately $94,500.

#### 4.2.2 Assembly

Assuming that each of us can assemble 15 tag units per hour, assembly adds $3 ($45/15) to the unit cost. This changes the 5,000 lot unit price to $10.586 per unit and the total lot price to $52,930.

Assuming that each of us can assemble 3 receiver units per hour, assembly adds $15 ($45/15) to the unit cost. This changes the 100 lot unit price to $74.954 per unit and the total lot price to $7,495.

#### 4.3 Grand Total

The grand total cost to research, develop, order parts, and assemble a 5,000 lot of tag units and a 100 lot of receiver units is $94,500 + $52,930 + $7,495, which comes out to a total of $154,925.
5 Schedule

Figure 5.1: Schedule

Tasks in blue are to be completed by everyone.
Tasks in red are assigned to Michael.
Tasks in yellow are assigned to Yue.
Tasks in purple are assigned to Cain.
Pale tasks are incomplete and dark tasks are complete.
6 Ethical Considerations


1) to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

   This ethic guideline is important to us because what we are working on is very tightly related to public health. Catching sick cows before they are used for meat or milk produce is an important part of our society.

3) to be honest and realistic in stating claims or estimates based on available data;

   It is important for us to not make any false or inaccurate claims in the design, research and testing that we do for our project. These will only hurt us and other engineers around us.

6) to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

   This project is aimed at improving all of our technical competence and knowledge to better our understanding of our field.

7) to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

   Throughout our design process we will be asking for honest criticism of our work by peers, TAs and professors. We must be able to accept criticism of our work even if we do not agree. Looking at our project from multiple points of view will help in the growth of our technical and business skills.

10) to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

   During the duration of the design process it is very important for us to assist and support each other in our work. We are here to improve our technical and general engineering skills and we can do that better by working together as a team.

6.2 Safety Guidelines

We adhered to strict guidelines. We had all values double checked, verified that all vendors sourced their materials legally, that all planned construction and machine shop work was to be done by certified professionals. On top of this, we repeatedly accepted criticism and implemented changes. All group members contributed equally and were not subject to any form of discrimination including but not limited to on the basis of sex, orientation, age, national orientation, or religion.
6.3 IACUC Policy [12]

- All research, teaching, and outreach activities at the University of Illinois involving vertebrate animals must be approved by the Institutional Animal Care and Use Committee (IACUC) before the activity begins.

- Once the activity begins, any proposed changes must be submitted to the IACUC as a major or minor amendment to the protocol. Changes must be reviewed and approved by the IACUC before they are implemented.

- The approval is for a 3-year period; the IACUC conducts mandatory annual reviews.

- If the approved use of animals is expected to continue beyond three years, a new Proposal for Activities Involving Animals must be submitted to the IACUC, reviewed, and approved before the active protocol expires.
References


Appendix A  Table of Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym / Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RTC</td>
<td>Real-time Clock</td>
</tr>
<tr>
<td>Rx</td>
<td>Receive / Receiver</td>
</tr>
<tr>
<td>Tx</td>
<td>Transmit / Transmitter</td>
</tr>
</tbody>
</table>
Appendix B  Circuit Schematic and PCB Drawing

Figure 1: Circuit Schematic

Figure 2: PCB Drawing