



ECE 4760: Final Projects

Thermistor Respiratory Monitor

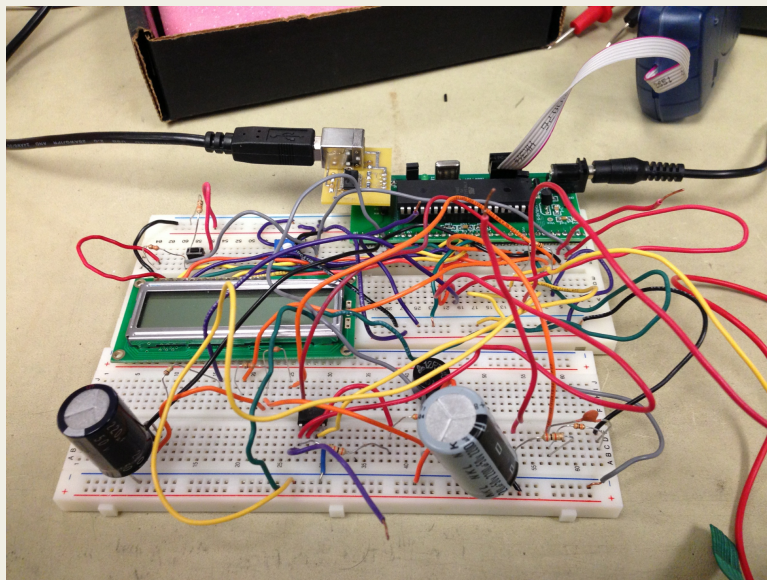
A low-cost, easy-to-use device to monitor the breathing of patients.

Maneesh Gupta (mg573@cornell.edu)

Hana Qudsi (htq2@cornell.edu)

Our final project for ECE 4760 is a respiratory monitor that was designed for low-resource environments. The device calculates a patient's breathing rate by detecting changes in temperature when the patient breathes through a mask. Features of the device include an alarm through a piezoelectric speaker which goes off when the patient stops breathing, and a low-battery indicator signal when the battery powering the device reaches a threshold voltage.

"This portable device can be used anywhere to give an alert when a patient stops breathing"



Prototype of our Thermistor Respiratory Monitor

Project Overview [top](#)

Motivation

For our project, we wanted to create a solution to a real-world problem. Many biomedical devices, such as commercially available respiratory monitors, are designed for the developed world and require a stable power-supply to operate. We wanted to implement a solution that is adaptable to different environments.

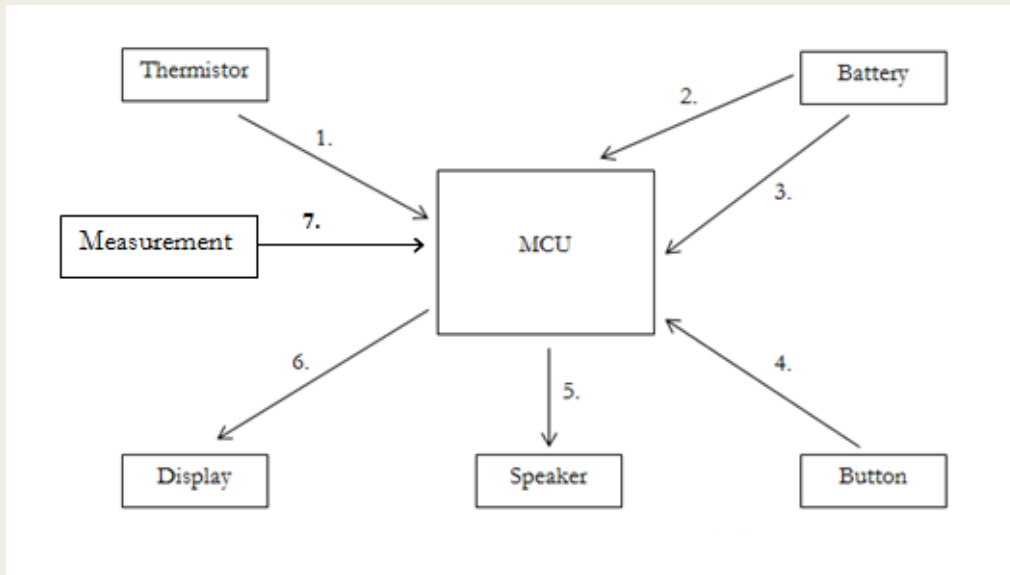
Our device uses several concepts we learned not only this semester in ECE 4760, but also in other classes at Cornell. We use analog-to-digital conversions to sample readings from both the thermistor and the battery, pulse-width modulation to generate a signal for the speaker, and timers to switch between tasks. Our implementation also includes various analog circuitry for voltage regulation and signal amplification. Developing a robust and accurate respiratory monitor served to be a challenging, yet rewarding experience.

Previous Work

When doing research for this project, we came across several respiratory monitors using different sensors for measuring breathing. Other methods of monitoring respiratory rate include a chest force sensor (to detect force produced by chest movement), impedance pneumograph (which uses skin electrodes to calculate transthoracic impedance), and pulse oximetry (which measures O₂ saturation in blood). While commercial devices to monitor respiratory rate exist, they use motion or force sensors to determine respiration rate. These monitors tend to be better suited for developed countries where a stable power supply is not a concern. We want to design our device geared to the needs of undeveloped areas. Therefore, we aimed for our thermistor monitor to be comparatively cheaper, easier to use, as well as reliable.

We also discovered another thermistor respiratory monitor research project. The design of the device differed from ours, in that it included two thermistors; one that remains in the outside environment for reference and one that goes inside the nostril.

High-Level Block Diagram



1. Thermistor Measurements

The resistance over the thermistor drops when its surrounding temperature increases, and goes back down when the temperature decreases. The voltage also, accordingly drops when a person exhales and rises when a person inhales. We use an operational amplifier to make the changes in temperature more apparent. The output of the amplifier is read into the microcontroller's ADC channel 0.

2. Voltage Measurements

Under battery operation, our device takes a voltage sample across the battery about every minute. Since our device uses a 9V battery, a voltage divider is used to drop the voltage so that it is about 2.2V at PINA1 (ADC channel 1).

3. Power

Our device can be powered through either a standard AC power supply or a single 9V battery.

4. Turn On Display

To conserve power, the user must hold down a button to turn on the display to read the respiration rate. This prevents the user from leaving the display on by mistake and draining the battery.

5. Output Sound

There are two different alarms for our device generated by a piezoelectric speaker. The first is higher in pitch and alarms if the patient is not breathing. The second is a signal that the device is running on low battery. While the first alarm is a continuous sound, the low-battery alert is a one second long tone.

6. Respiration Rate Display

Breathing rate is measured and displayed in breaths/min on an LCD. This allows whoever is monitoring the patient to see if the patient is breathing too fast or too slowly.

7. Measurement

This task does the actual calculation of the respiration rate using samples from ADC0.

Design Decisions

Microcontroller

When choosing a microcontroller for this project, we first wanted to find the smallest microcontroller possible so that it could be mounted right onto the device's mask. However, when identifying how many I/O ports and ADC channels would be necessary for our project, we realized that the AtMega1284p microcontroller that we had been using all semester would be ideal.

Thermistor

During the initial stages of our project design we experimented with several thermistors and their placement on the device. Since a drastic temperature difference is not produced when a human breathes, we needed the smallest, and therefore most sensitive thermistor we could find. Larger thermistors we experimented with either could not detect a "breath" or had a very slow response time. We also thought a mask around the patient's nose and mouth would be the least intrusive placement of the thermistor.

Display

Because we wanted to design our device to be low-powered, choosing the right display for the respiratory rate was a concern. Originally, we planned to use two 7-segment displays which turned on when the user pressed a button. However, we decided to use an LCD screen for a nicer display. In order to reduce power usage, the LCD only receives power when the user presses the display button, which reduces the load on the battery.

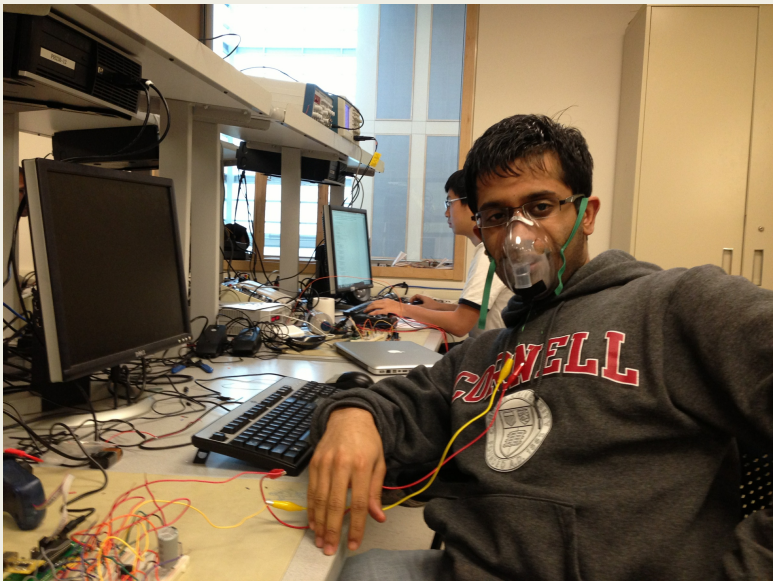
Standards

As a medical device, this respiratory monitor must meet regulatory requirements outlined by governments. This includes meeting HIPPA standards which protects individuals' medical records and other personal health information, including respiration rate. However, since we are not planning to store this information, this regulation should not be an issue. Before it can be sold commercially in the United States, the device would also need FDA approval.

Hardware Design & Implementation top

There are six different hardware components necessary to make the respiratory monitor work. The six components are listed below and are described in that order.

1. Thermistor Measurement
2. Thermistor Amplification
3. Analog Comparator
4. Capacitor Charger
5. LCD Display Button
6. Battery Monitor



Use of device

1. Thermistor Measurement

We used a thermistor based measuring system to measure the breathing rate of the patients. The thermistor is mounted inside a mask

which is worn by the patient. The mask is a standard nebulizer mask that is used by asthma patients. The thermistor is mounted inside the mask so that it is directly in front of the patient's mouth. As the patient breaths the hot air from the patient's breath changes the resistance (R_{th}) of the thermistor. As a result, the voltage across the thermistor (V_{th}) will also change proportionally to how the patient breaths. We can therefore use V_{th} as an indirect measurement of how the patient is breathing. To measure V_{th} we used a voltage divider as shown below in Figure 1.

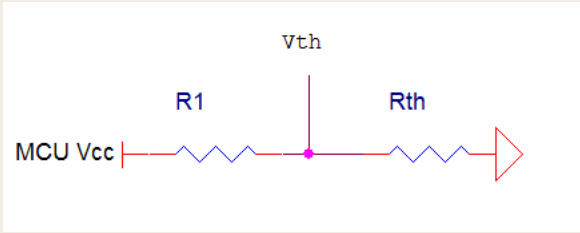


Figure 1: Thermistor Measurement Circuit

When the thermistor is at room temperature its resistance is approximately 1.2 k Ω so we choose R_1 to be 1 k Ω . This was chosen so that V_{th} would be about 2.5 V if no one was breathing on the thermistor. We wanted to have 2.5 V be our no breathing voltage so that extreme inhalations or exhalations could be detected with our device.

2. Thermistor Amplification

The thermistor we choose for our project was the NTHS0402 from Vishay Dale. This thermistor has a thermal time constant of 5 seconds which makes it extremely sensitive. During experimentation we found that the change in voltage from a person breathing on the thermistor was on the order of 10 mV. Since this was too small of a change to work with we decided to amplify this change by using an Op Amp with a high pass filter. An Op Amp alone will not suffice because it will amplify the entire signal whereas we only want to amplify the change in voltage. Literature research shows that the average adult male breaths about 15-20 breaths/min and the average infant breaths about 30-60 breaths/min. Our slowest desired period is about 4 seconds. We choose the time constant for our high pass filter to be 22 seconds. This means that any signal with a period less than 22 seconds will be amplified by our op amp. We choose 22 seconds so that our device could detect hyperventilation as well as normal breathing. The time constant was set to 22 seconds by choosing R_2 to be 10 k Ω and C_1 to 2200 μ F as seen in Figure 2. The gain was then set to 4 by choosing R_3 to be 30 k Ω .

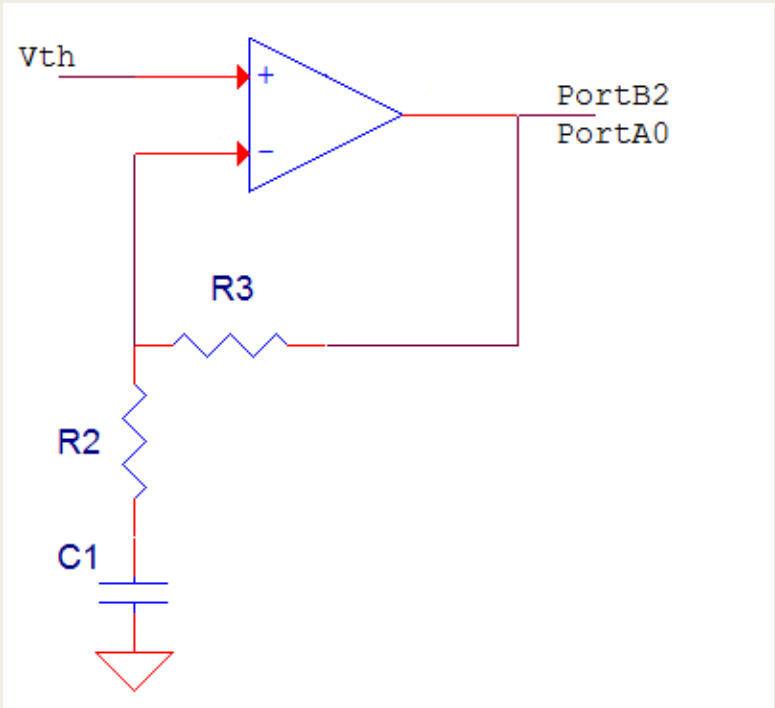


Figure 2: Thermistor Amplification Circuit

The output of this circuit goes to both PortA0 and PortB2. PortA0 is the input to the ADC while PortB2 is the + reference pin for the AC (see software design section).

3. Analog Comparator

The respiratory rate is measured by using an analog comparator (AC) to compare the amplified voltage across the thermistor to a

reference voltage. Since the respiratory signal is approximately equivalent to a sine wave with a DC offset, we choose the reference voltage to be at the DC offset. However, patients of different ages will have different steady state breathing temperatures. For example older patients will breath out more hot air than younger patients. This means that the DC offset will be lower than those of younger patients. To automatically determine the reference voltage for a patient we used a low pass filter shown below in Figure 3. This low pass filter has a time constant of 22 seconds which means it rejects all of the signals we are interested in. Since it rejects the respiratory signal the output of this filter must be at the average value of the incoming signal. As the incoming signal is equivalent to a sine wave with a DC offset, the average value of the signal is the DC. The output of this filter than goes into the voltage reference pin of the AC as shown in Figure 3. The respiratory signal comes from PortB2 and goes into both the filter and positive voltage pin of the AC of the MCU. To achieve a time constant of 22 seconds R4 is 100 k Ω and C2 is 220 μ F.

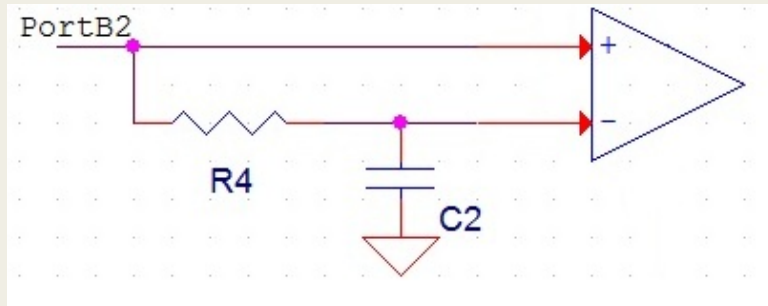


Figure 3: Analog Comparator Circuit

4. Capacitor Charger

The time constant for the thermistor amplification circuit described above is 22 seconds which means it takes several minutes for the capacitor to charge up. During this charging period the device does not work properly because the capacitor is not fully charged. To decrease this charging time another circuit was added with a smaller resistor so that the time constant for charging would be faster. The charging capacitor schematic is shown below in Figure 4. When PortB4 and PortB5 are set to 5 V and 0 V respectively (see software design for further details) the capacitor begins to charge through R6. Both R5 and R6 are 330 Ω so that they would provide as little resistance as possible. The time constant for charging then becomes 0.73 seconds and the total time for charging is about 6 seconds. R5 is necessary to provide protect for PortB4 so that it is not damaged.

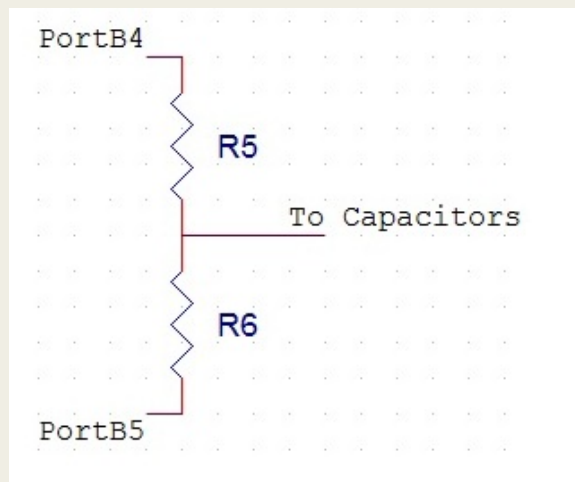


Figure 4: Charging Capacitor Circuit

5. LCD Display Button

In order to conserve power the user must hold down a button to turn the LCD on. This way the LCD is not on when it is not being used and it cannot be left on by mistake. This was accomplished by connecting the LCD Vcc pin to the MCU Vcc via a push button as shown in Figure 5. When the button is pushed the LCD Vcc pin receives 5 V. Resistor 10 and Rresistor 7 are needed to ensure that the voltage at the pink node is below the threshold voltage of the MCU pins. This way PortB1 will read 0 V when the button isn't pushed and 5 V when the button is pushed. Both R10 and R7 are 330 Ω .

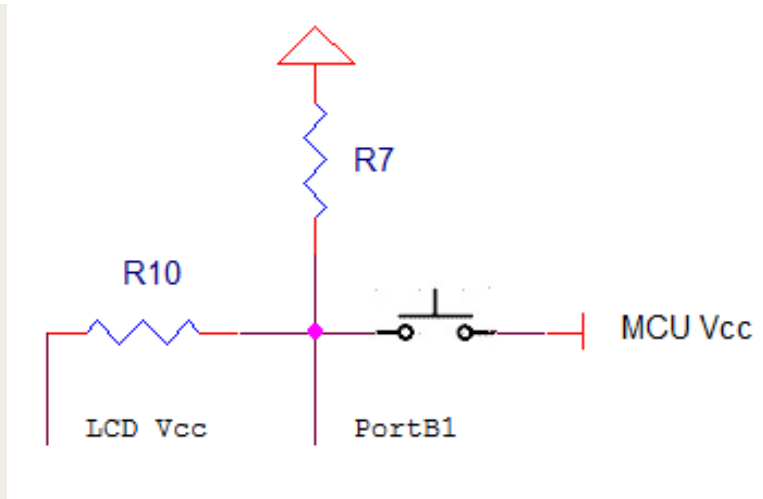


Figure 5: LCD Display Button Circuit

The LCD we used the MDL(s)-16264 LCD from Vartronix along with a 10 k Ω trimpot to control the contrast.

6. Battery Monitor

To measure the voltage of the 9 V battery we used a voltage divider along with a low pass filter as shown in Figure 6 below. The voltage divider is used to convert the voltage range from 0-9 V to 0-5 V so that the MCU will not be damaged. The ADC is then used convert the analog voltage to a digital format so that the MCU can analyze it. The purpose of the low pass filter is to eliminate any high frequency signals that might occur. The time constant of the filter is 1 μ s and is achieved by setting C2 to 1 nF and R8 to 1 k Ω . R9 is set to 3 k Ω so that the voltage divider outputs $\frac{1}{4}$ of the original voltage.

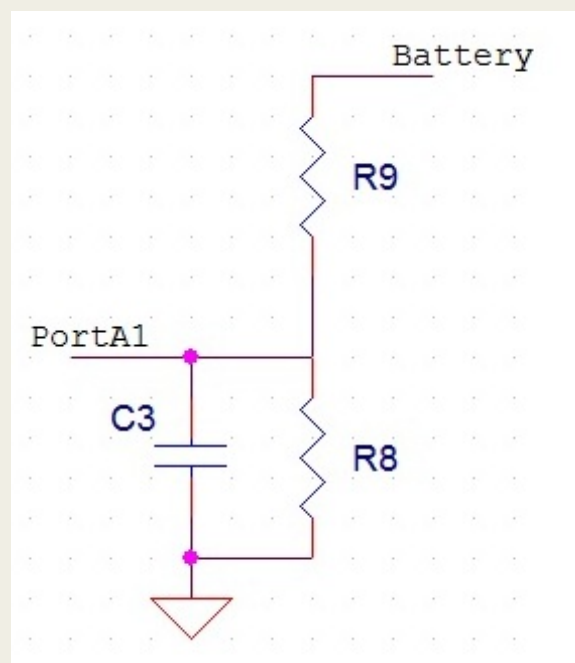


Figure 6: Battery Measurement Circuit

Software Design & Implementation [top](#)

Software Overview

The C programming language was used to program the MCU using the AVR Studio version 4.15 and was compiled using the WINAVR GCC C compiler. The software for the respiratory monitor contains 5 different tasks and 2 interrupt service routines (ISR). The first ISR is the Timer0 Compare Match Vector. This ISR activates every 16.4 ms and decrements three different timeout counters whenever it activates. It also zeroes the variable count (used to measure respiration rate) to prevent it from overflowing. Since the clock of the MCU is 16 MHz, we set the prescaler to 1024 and OCR0A to 255. Setting OCR0A to 255 means the ISR will activate every 256 clk cycles. The

second ISR used was the AC ISR. This ISR activates when the onboard AC outputs a 1. This ISR was used to measure the respiration rate of the person. This will be described in detail below in the appropriate section.

Main

The main function initializes the device and then goes into the main loop. Both the low pass and high pass filters have very large time constants (22 seconds) and capacitors. This means that they will take an extremely long time to charge before the overall device can begin to work. This start up time was on the order of minutes, which was too long in our opinion. To solve this problem we created a separate circuit that would charge our capacitors using much smaller time constants. This was done in the first 10 lines of code in main before any other function was initialized. First, PinB5 and B4 are set to output. B5 is set to 0 V and B4 is set to 5 V. This essentially turns on the Vcc and ground for the charging capacitor circuit. The charging capacitor circuit is left on for 5 seconds before it is turned off. It is turned off by switching B4 and B5 to inputs. Afterwards the initialize functions are turned on and the while(1) loop begins.

The while-loop goes through a series of conditions to see if the device needs to call one of the tasks. The thermistor and sampling tasks are both timed, so if either timer count equals 0, the counter is reset and the task is called. The thermistor sampling task is called every second, while the battery sampling task is called every minute. The measure task is called when the measure_flag indicates that a measurement is ready to be taken. Lastly, the display task activates when the user holds down the display button.

Thermistor Sampling

Samples from the thermistor are taken through PINA0. First, the ADMUX register is set to turn on the left adjust result and external reference. The MCU then waits for a conversion to occur, and then sets the variable sample_old to the current sample and the sample variable to the high bits from the ADC. The absolute value of the difference between sample_old and sample is used to determine if the patient has stopped breathing. If this difference is less than or equal to 8, then this is considered a "no breath". If the MCU detects 10 consecutive "no breaths", a PWM signal is generated to produce a sound from the speaker. If the difference is above the tolerance, then the "no breath" counter variable c is reset to 0 and the PWM signal is set to be turned off.

The tolerance for the difference between breaths was found through experimentation. Looking at the ADC values on the UART, the range of the sampled values was 6-8 when a person was not breathing. To compensate for this fluctuation, we compare the difference between consecutive thermistor samples against 8 instead of 0. We also decided to have a counter to keep track of the number of "no breaths" detected because depending on how a person is breathing, consecutive samples could fall into the tolerance range. Counting up to 10 consecutive "no breaths" drastically reduces the number of false positive alarms.

ADC sampling and PWM signal generation were concepts we learned earlier in the semester in Labs 2 and 3.

Battery Sampling

To take a sample from the battery, ADC channel 1 on PINA1 is turned on by setting the ADMUX register. The MCU waits for a conversion to occur and then sets the variable bat_samp to the high bits from the ADC. The sample is then compared to the ADC value that was found to correspond to a voltage of 7.6 V. During testing it was found that device operation becomes unstable at 7.5 V. Therefore, we decided to warn the user when the voltage of the battery becomes 7.6 V

If the battery sample is found to be less than or equal to the tolerance, then a PWM signal is generated, left on for about 1 second, and then is turned off. The PWM channel which produces a tone for a low-battery warning has a larger prescaler than the PWM channel to produce the alarm for when a patient is not breathing, making the low-battery tone lower in pitch.

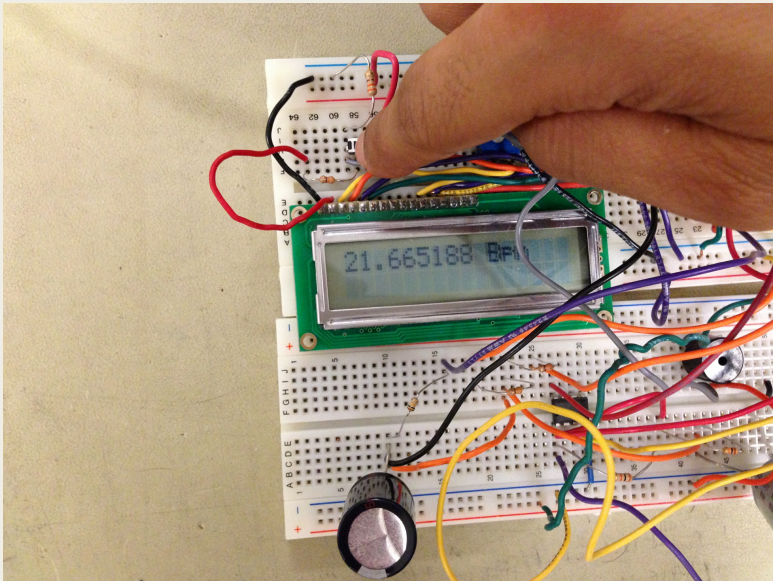
Measuring Task

The respiratory rate is measured by using the AC and measuring the time difference between when the voltage across the thermistor crosses the reference voltage. The first part of this is done in the ISR. When the voltage across the thermistor crosses the reference voltage the AC ISR activates. It then stores the number of clk cycles into the variable tau2 and stores the number of clk cycles of the previous crossing in tau1. The difference between tau2 and tau1 is then stored in the variable period. In essence this ISR measures the difference in the number of clk cycles between two reference voltage crossings. Once this measurement is done the measure flag is set and the measure task is activated. In the measure task an average difference in clk cycles is calculated by doing a weighted average between the newly measured difference in clk cycles and the previous average difference in clk cycles. The weights are 0.75 of the previous average and 0.25 of the new difference. This weighted average is taken so that an average respiratory rate can be calculated. This average difference in clk cycles is then converted to respiratory rate in breaths per a min (bpm). This is done by converting the difference in clk cycles to minutes and taking the reciprocal. We take the reciprocal because the AC ISR is activated every breath, i.e. it measures the time for one breath.

While writing this code we ran into the problem that the AC ISR was activating several times per one crossing. This was happening because as our signal crossed the reference voltage the noise in our system caused the signal to cross the reference voltage several more times. To solve this when the AC ISR is activated the first time the first line in the ISR turns off the AC ISR and the flag called flag is set. Then in main there is an if statement that says if flag is set to 1, wait 10 ms, turn on the AC ISR and reset flag. The 10 ms delay is used to wait for our signal to cross the reference voltage far enough so that there is no interference from the noise.

Display Task

The display function is used to display the respiratory rate on the LCD when the user is holding down the LCD push button. This was achieved by having the display function activate when PinB1 was read as 1. When PinB1 is 1 the LCD is initialized and the value of the respiratory rate stored in the variable breathrate and written to the LCD buffer. This value was then displayed on the LCD.



Display of respiration rate

External Project Code

Author	Description
Scienceprog.com	LCD library-used for displaying the respiration rate

File Breakdown

Name	Description
resp_monitor.c	ATmega1284p implemenation of thermistor respiratory monitor
lcd_lib.c	ATmega1284p source code for display onto an LCD
lcd_lib.h	ATmega1284p header for display onto an LCD

Conclusions [top](#)

Results

The major results of our design are as follows:

- Measures respiratory rate with error of less than 10%
- Takes less than 5 seconds to determine if patient isn't breathing
- Startup time is about 30 seconds
- LCD only receives power when the LCD button is held down
- Piezoelectric speaker is activated when patient isn't breathing or the battery is below 7.5 V. Different sounds are used for not breathing vs low battery
- Speaker is automatically turned off when the patient begins to breathe again
- Runs off of 9 V battery

Final Thoughts

Our device met the original design specifications we set out to accomplish at the beginning of this project. We were able to accurately

measure the respiratory rate, determine if someone was breathing or not and output an alarm if he or she wasn't, measure a battery and output an alarm if it was too low and turn on the LCD only when a button was being held down. We were surprised that we were able to get the piezoelectric speaker to go off in under 5 seconds when we stopped breathing into the mask. This is an extremely important accomplishment because it is imperative to know as soon as possible when a patient is not breathing. We were also surprised by how much lag there was in our system. At first our device took several minutes to starting amplifying the voltage across the thermistor but once we determined the cause we were able to get that down to under 5 seconds. Unfortunately there is still some lag between when the voltage across the thermistor begins to be amplified and when the device begins measuring respiratory rate. Fortunately this lag is very small so it is still acceptable but in our next iteration of this design we would like to minimize this. We believe it has to do with the low pass filter that creates the reference voltage for the device but this theory needs to be tested. We would also like to increase the accuracy of the measured respiratory rate. This can be done by either fine tuning the weights or by using a different averaging method. We don't think the current method is the best one because it doesn't coverage to a final value fast enough and it isn't sensitive enough to change. Satisfying both of these requirements is difficult to accomplish since satisfying one generally means making the other worse. Hence, more research into appropriate averaging methods is needed. We would also like to do a more in depth power analysis so that we can minimize the power requirements further. We saved power by only turning on the LCD when the button was held down but more can be done. For example, using the AC instead of the ADC to determine if someone is breathing would be save power.

Safety

To enforce safety in our design we choose to use a nebulizer mask which is a standard mask used by healthcare professions all over the world. This mask is easy to put on, comfortable to wear and prevents infants from picking at the thermistor inside. The other dangerous part of our device are the big capacitors. However this danger can easily be avoided by using smaller capacitors and increasing the resistance of the corresponding resistors so that the time constants are still met.

Interference

The main form of interference that can affect our device is temperature interference from the ambient environment. To prevent this interference we enclosed our thermistor inside a nebulizer mask so that it would not be exposed to the external environment. We also choose a nebulizer mask because this mask has holes in it to allow cross ventilation. This cross ventilation is needed so that the thermistor is not saturated by the patient's breath.

Usability

Our intention with this project was build a respiratory monitor designed to be simple to use. The only information displayed is a number so there are no language barriers that need to be considered. The instructions for the device are very simple. Place the mask on the patient, then turn on the device, and wait 1 minute to ensure device has started properly. Afterwards, the LCD button may be held down to display the respiration rate. If the patient isn't breathing an alarm will sound, or if the battery is low another alarm will sound.

Conforming to Standards

Since our device was designed for the developing world, there isn't a set of standards that we could compare our device too. In general we designed our device so that it would be safe and easy to use. We choose to use a nebulizer mask because it is widely used and trusted by healthcare professionals as well as cheap. We also made sure there were only numbers on the device so that there would be no language barriers. Furthermore, we sought to make the device power efficient for use in low resource environments

Intellectual Property Considerations

There are many other medical devices that use thermistors to monitor the human body, mostly for the purpose of sleep apnea. The main feature that distinguishes our device from these commercial products is that our device uses one thermistor while these other devices have two or more thermistors. Even with this distinction we don't believe there are many opportunities for patenting with our device. However we do believe there are publishing opportunities for our device which we plan to pursue. For example, we intend to submit an abstract to the American Society for Artificial Internal Organs (ASAIO) student design competition held this June 12-15. This competition is intended to promote undergraduate engineering design in medical technology. We feel that our project is a good fit for this competition.

Ethical Considerations

Throughout the last few weeks while we worked on or device we made sure to adhere to the IEEE Code of Ethics. In writing this report we did not embellish the results of our project and we gave over estimations for the error our device produced. We also described the strengths and weaknesses of our device to ensure that whoever read our report would understand what our device can and cannot do. In addition, while we worked in lab we helped our fellow students with debugging their projects. For example, another group came to us asking for help with their AC. They had the exact same problem we had and we were able to save them countless hours debugging by pointing out their problem and telling them the solution. Furthermore, we designed our product so that it would not only work but it would

also be safe at the same time. We choose to buy a nebulizer mask instead of modifying some mask we had laying around because nebulizer masks are approved medical equipment used by healthcare professionals that are meant to be worn on the face. We also pointed out the potential danger with having large capacitors in our device and constructed a solution for this problem. In summary, we followed the IEEE code of ethics by accurately portraying our results, assisting our colleagues, choosing safe parts and pointing out dangers of our device.

Legal Considerations

Because this device is desired for the third world there are not many legal considerations we need to take. If this device were to be sold in the U.S. it would need to pass FDA and we would have to make sure we don't infringe on any patents which we mostly likely do. Furthermore, our device would be classified as a Class 1 medical device due to its simplicity. If we wanted to get this device on the market our best path would be to file a 510(k). What we would need to do is prove that our device is "substantially equivalent" to an existing device already approved by the FDA. This shouldn't be too difficult because there are existing medical devices using thermistors

Appendices [top](#)

A. Parts List and Costs

Category	Item	Vendor	Unit Cost	Quantity	Total Cost
Electronics	White Board	ECE 4760 Lab	\$6.00	2	\$12.00
	AtMega1284p Target Board	ECE 4760 Lab	\$12.00	1	\$12.00
	9V Battery	Anywhere	\$2.00	1	\$2.00
	LCD	Varitronix	\$8.00	1	\$8.00
	Nebulizer Mask	justnebulizers	\$5.00	1	\$5.00
	NTC Thermistor SMD 402	ECE 4760 Lab	\$0.00	1	\$0.00
	10K Trimpot	ECE 4760 Lab	\$0.00	1	\$0.00
	Piezo Electric Speaker	ECE 4760 Lab	\$0.00	1	\$0.00
	Wire	ECE 4760 Lab	\$0.00	N.A.	\$0.00
	Button	ECE 4760 Lab	\$0.00	1	\$0.00
	Op Amp LF353N	ECE 4760 Lab	\$0.00	1	\$0.00
Resistors and Capacitors	Through-Hole 330Ω Resistor	ECE 4760 Lab	\$0.00	4	\$0.00
	Through-Hole 1kΩ Resistor	ECE 4760 Lab	\$0.00	2	\$0.00
	Through-Hole 3kΩ Resistor	ECE 4760 Lab	\$0.00	1	\$0.00
	Through-Hole 10kΩ Resistor	ECE 4760 Lab	\$0.00	1	\$0.00
	Through-Hole 30kΩ Resistor	ECE 4760 Lab	\$0.00	1	\$0.00
	Through-Hole 100kΩ Resistor	ECE 4760 Lab	\$0.00	1	\$0.00
	Through Hole 220μF Capacitor	ECE 4760 Lab	\$0.00	1	\$0.00
	Through Hole 2200μF Capacitor	ECE 4760 Lab	\$0.00	1	\$0.00
	Through Hole 1nF Capacitor	ECE 4760 Lab	\$0.00	1	\$0.00
				Total	\$39.00

B. Source Code

Main file: [resp_monitor.c](#)

LCD header file: [lcd_lib.h](#)

LCD source code: [lcd_lib.c](#)

C. Specific Task Breakdown

Maneesh Gupta	Hana Qudsi
Measure Task & Debugging	Thermistor Task & Debugging
Display Task & Debugging	Battery Task & Debugging
AC ISR	Timer0 ISR
LCD Button Circuit	Initialize Code
Initial Capacitor Charging	Website Design
Design Report	Design Report

References [top](#)

This section provides links to external documents, code, and websites referenced and used throughout the project.

Datasheets

- [Thermistor](#)
- [Operational Amplifier](#)
- [LCD Screen](#)
- [AtMega1284p](#)

Vendors

- [Nebulizer Mask](#)

Background Info

- [Analog Sampling](#)
- [Thermistors](#)
- [Respiratory Monitors](#)

References

- [IEEE Code of Ethics](#)
- [Template Website](#)
- [Header for LCD library](#)
- [Source code for LCD library](#)

Acknowledgements [top](#)

We would like to thank Professor Bruce Land for encouragement and inspiration. Spending hours of fine-tuning this project was a memorable life experience.

Our gratitude also goes to Amrit Singh and Mike Kilzer, our TAs, who helped us through various issues with our project.