Team 17

Facilitator: Dr. Mukkamalla

Ian Bacus, Anish Gill, Yousef Gtat, Yujie Hao, Anna Little, Craig Stoddard

Executive summary

<inst>

Table of contents

<inst>

Introduction

This report describes a smart health monitoring device for use in personal daily health and hospital nurseries. A data acquisition circuit housing an infrared thermopile is described, as well as software components for transmitting, correcting, and displaying temperature measurements to users. The design is intended to be an external sensor package for smartphones which allow them to take noncontact temperature readings of human subjects and obtain temperature measurements. By using a smartphone as the foundation of the device, network connectivity and device recharging are taken care of. The method employed by this device is advantageous to traditional contact based temperature sensors. The design provides a faster way of collecting data than traditional means, and it allows patient information to be analyzed programmatically. For personal daily health, this design could be extended to allow personal data to be recorded in the cloud system, where it is easier for checking in the future. In a hospital setting, this results in less training being required for employees, and less time spent testing each patient for problems.

Background

Not many advances in temperature measuring technology have been made in the medical field since the invention of mercury thermometer. There is a common infrared temperature measurement gun found in industrial settings, but these devices can be inaccurate and are not intended to be used for body temperature. In today's medical facilities the instant in ear thermometer is the most common temperature measuring device, however, this is a contact based sensor and does not utilize a means to store temperature results and requires cleaning. There are models such as the ‘Mendipweather’ which utilize the iphone circuitry but is still contact based as well. The design goal is to create a fast measure taking device that does not involve contact with the potentially sick patient, and create a storage space for past testing results.

With the help from the University of Michigan, a key component and concept called ‘Hijack’ allows for the the iphone to provide power to a peripheral device as well as a means of communication using the iphone’s built in audio jack. Combining this circuitry with the TMP007 temperature sensor will allow utilization of the phone’s auxiliary jack to power up peripheral devices.

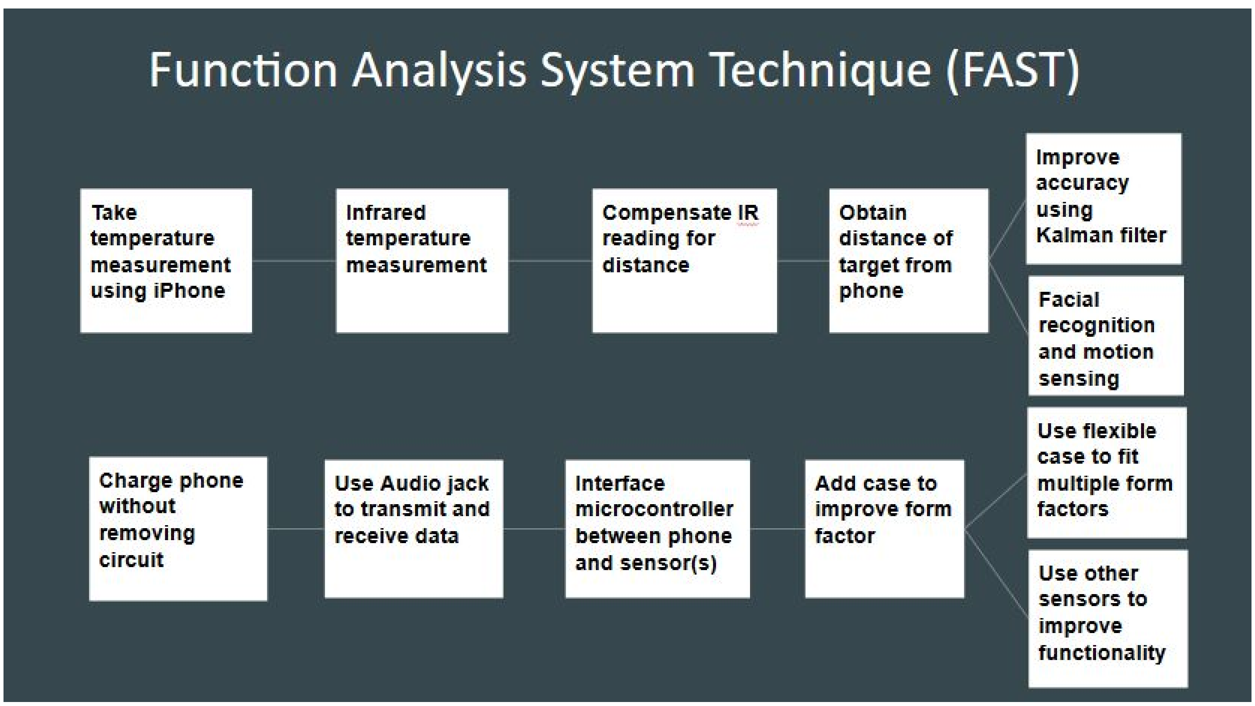
Also, useful work in distance compensation of infrared sensors has been carried out by Yildizyan et al. This, along with a focusing lense, will potentially allow the design to be used from a distance greater than the 3.5cm the TMP007 is slated to be accurate for.

Objectives

The project applications mainly focus on Personal Health and Wellness, and improving medical tools for Hospital Clinics. Personal health is a health record where health data and information related to the care of a patient is maintained by the patient. The targeted customers in this category are those who seek to live a healthy life and run occasional checkups on themselves for early diagnosis and treatment. Keeping a personal record of the patient also aid with providing a complete and accurate summary of an individual's medical history. Personal health devices are growing field nowadays considering the successful selling value of such devices as Fitbit or Apple Watch. Data from these electronic devices can be collected passively from a smartphone through the data, and shared with medical institutions and hospitals to be accessible online. This grant patients access to a wide range of health information sources, best medical practices and health knowledge. All of an individual’s medical records are stored in one place instead of paper-based files in various doctors’ offices. Upon encountering a medical condition, a patient’s health information is only a few clicks away, which helps clinicians and doctors to make better treatment decisions by accessing continuous data throughout the day, week, month, or even years.

The goal for the final product, THERM, is to be utilized as medical tool for Hospital Clinics. THERM should be compatible to work with an iPhone application to conduct temperature reading within seconds. The device is attachable to the back of the iPhone, yet small enough to fit in the user's pocket. This provides doctors, nurses, and health inspectors the flexibility to take temperature measurements of their patients quickly thus replacing the old medical temperature toolkit in hospitals. THERM measures temperatures with an accuracy of ±0.1F degrees within an ambient temperature range of 60F to 90F degrees, given that the normal room temperature is approximately 70F. THERM incorporates several sensors from the iPhone such as the camera to take temperature measurements from a distance of 2 feet of the target. By utilizing the iPhone sensors, THERM manufacturing cost drops to under $40 per chip.

<inst> what is this section?

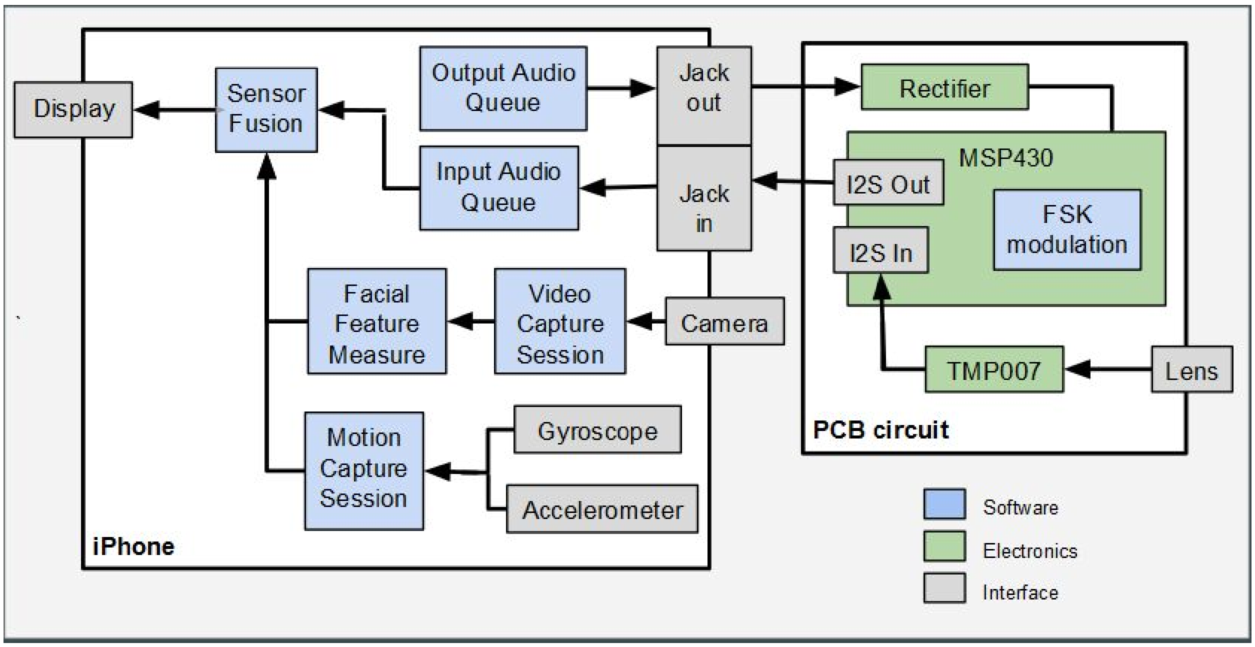


Conceptual Designs

The proposed project required that the design solution would attach to a smartphone. The majority of smartphones currently are either android or iPhone and their functionality is almost identical. The decision was to use an iPhone for the implementation because one was already available to use. Since smartphones have limited I/O the interface choices were USB or audio jack. Both choices were considered, however, as shown in figure --- being able to charge the phone without removing the temperature sensor was a key part of the design, thus the audio jack was chosen.

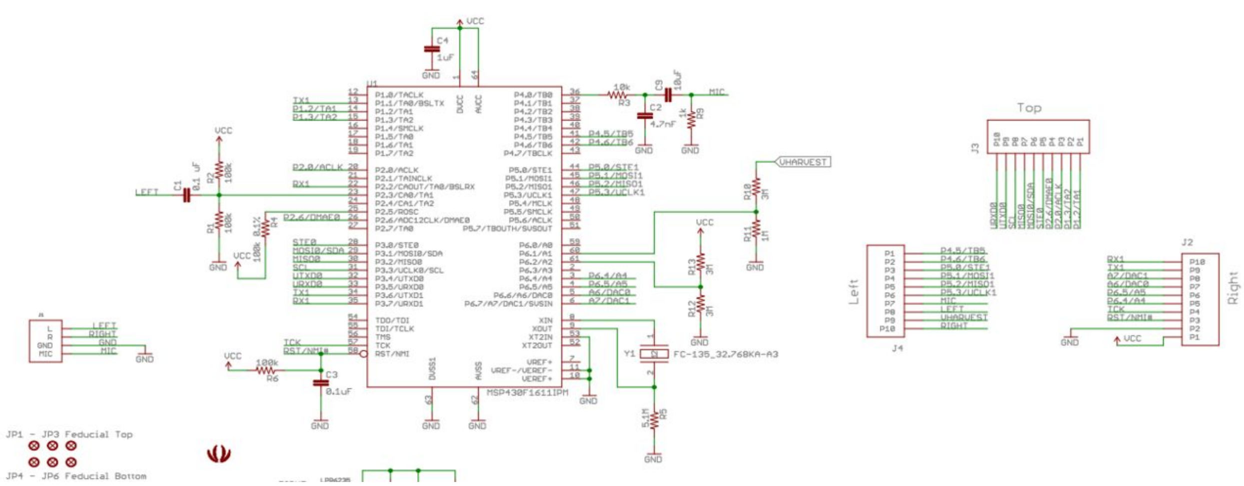
A.

Proposed design



<u>Hardware</u>

The hardware implementation consists of creating a Printed Circuit Board (PCB) using the schematic from the ‘hijack’ paper as seen below. The PCB will be the link between the iPhone and the MSP430 and will also contain the temperature sensor TMP007. However, the existing schematic from the ‘hijack’ paper will need to be modified to include the TMP007. The PCB needs to be designed compact enough so different sized phones will be able to use the external circuit.



The TMP007 is a thermopile sensor able to measure the temperature of an object without direct contact. It absorbs infrared energy from objects at wavelengths between 4 micrometers and 16 micrometers, using a special window that acts as a bandpass filter for light. It also contains an internal math engine to calculate the target temperature, as well as a non-volatile memory for storing calibration settings. The sensor package is 1.9 millimeters by 1.9 millimeters.

A focusing lens will be implemented to reduce the field of vision of the temperature sensing unit, thereby increasing the distance from which an accurate temperature reading can be taken. The unmodified TMP007 has a 110° field of view, restricting its maximum range for measuring temperature from an average human face (modeled as a 9cm*9cm circle) to 6.3 cm.

110° / 2 = 55°

facial width * tan(55°) = maximum measuring range

9cm * tan(55°) = 6.3cm

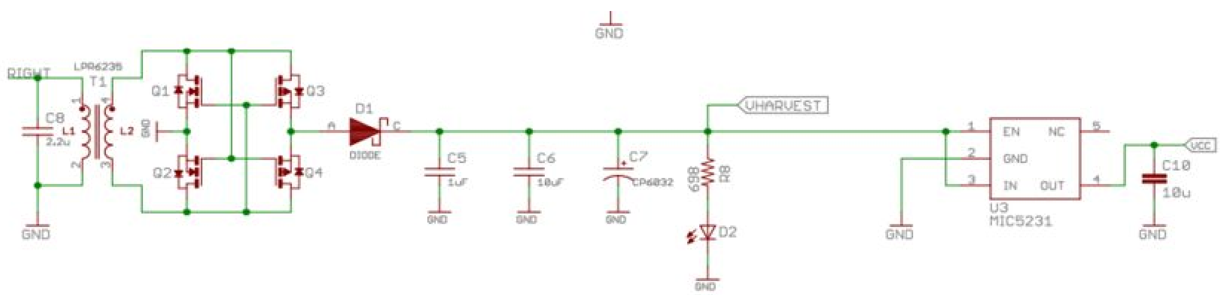
To achieve a range of 0.75m, meeting the objective, the field of view must be narrowed to any value lower than 83°.

Field of view = arctan((maximum range) / (facial width))

83° = arctan(75cm/9cm)

The PCB acts as the communication interface between the MSP430, TMP007, and the iPhone audio jack. There will be two channels coming from the audio jack into the circuit; these are the left and right channels. There will only be one channel of output from the circuit back into the iPhone; this is the microphone channel. The left channel will be used to draw power from the iPhone to power the MSP430, and the right channel and microphone channel will be used for transmitting and receiving data. Therefore, the PC headset is the audio jack being used.

The circuit will receive power from the left channel of the audio jack. An AC diode rectifier will be used to convert AC voltage to DC voltage that can be used to power the microcontroller. This rectifier is included in a circuit schematic from the ‘hijack’ paper as seen below.



Software

The software implementation consists of an Objective C iOS application, and microcontroller firmware in C. The iPhone acts as the power supply, main computational area, and user interface for the temperature sensing device. The microcontroller resides on the PCB described in the previous section, and its responsibility is to receive information from the TMP007 thermopile and transmit it to the iPhone.

The microcontroller application sends and receives data over I2C compliant interfaces, and provides a clock signal for the temperature sensor chip’s I2C interface. The design does not currently utilize other sensors than the temperature sensor, so I2C is only being used to receive information from the TMP007 temperature sensor and to transmit information to the audio jack. This signal is a bit stream with voltage levels of 0V and 3.3V for zero and one bits. The microcontroller code receives a 9600 baud signal from the temperature sensor consisting of 32 bit serial samples, and these are forwarded out unchanged through the transmission portion of the I2C interface. This process is interrupt driven and initiated by the temperature sensor, which sends samples to the microcontroller at 9600 baud. The microcontroller takes each byte as it arrives and passes it out at the same rate to the microphone output, which is sent to the iPhone for ADC conversion and data collection.

Future ideas

- In the design proposed by Kuo et. al, frequency shift keyed modulation is implemented in the microcontroller to allow communication from multiple input devices over the limited audio band of the audio interface.
- Turn the microcontroller into a low power mode based on commands received from the iOS application, so that it only reads and sends temperature readings when facial measurements are detected
-

The iOS application consists of three main modules: for audio control, video capture with facial recognition, and motion sensing. Each of these applications runs one or two threads associated with receiving data from the associated hardware sensors, respectively the audio jack codecs, camera image buffer, and gyroscope and accelerometer sample buffers. All three of these modules are essentially sources of real time data that is used to generate a facial temperature of a human target. The audio interface is used to obtain temperature measurements sent by the microcontroller application, and the data from other sensors are used to to provide real-time compensations of the acquired temperature readings. The audio module also works as a destination for data, as the microcontroller is capable of receiving commands over the audio jack. Additionally, this audio module provides the 18kHz oscillating signal that is rectified to generate power for the components on the PCB circuit.

Audio interface description

The audio interface consists of a playback thread and a recording thread of execution, both of which are callback functions invoked by system events in response to audio buffers changing state.

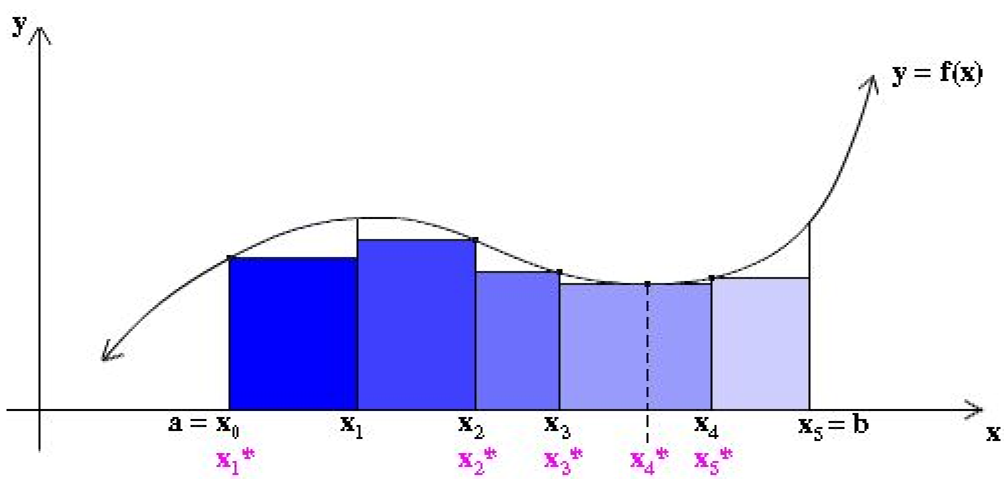
For the playback system, freshly emptied buffers are refilled with new linear PCM samples to be played back. For the playback function, a sinusoid is generated one sample at a time and delivered as a mono signal to the default audio output interface, determined by the presence (or lack thereof) of an audio jack connector in the audio jack port. The generated sinusoid has a frequency of 18 kHz and an amplitude of 500mV. The signal is generated with a sampling frequency of 44.1 kHz to respect the Nyquist rate, where each sample goes into the cell of a buffer that is converted to audio using the linear PCM system.

The recording portion obtains samples at 38400 Hz, four times the baud rate used to transmit data over the microphone line. These samples also correspond to linear PCM data, such that each cell in the buffer of an input packet represents a discrete and digitized value of the input signal. On the arrival of every packet over the audio interface, the recording callback extracts each sample of the acquired signal and enqueues it into a data structure that acts as a triggering buffer. This triggering buffer is used to detect occurrences of start bits in the reconstructed signal, which allow multiple audio packets to be recombined into a single 32 bit sample of data as generated by the TMP007.

Motion sensing portion

The motion sensing module is configured to work with an internal gyroscope, accelerometer, and magnetometer to determine instantaneous attitude along three degrees of rotation (roll, pitch, yaw) and instantaneous acceleration along three degrees of motion (x, y, z) that are fixed relative to the phone. The objective of using this data is to measure displacement along the axis traced by the back-facing camera across two distinct timepoints. The axis of the back facing camera is defined as the vector orthogonal to the plane of the phone screen. Apple provides an API that extracts this information for axes fixed with the phone, such that the acceleration along the back facing camera's axis can be measured directly without adjustment by the attitude. The API also filters out the effects of gravity from accelerometer readings, so that "user acceleration" and acceleration due to gravity along the x,y, and z axes can be obtained independently. In this case, user acceleration in the z direction is the only necessary piece of information to determine depth along the axis of interest.

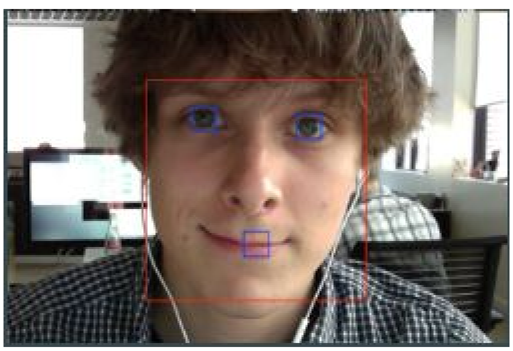
To obtain the distance travelled along this axis between two time points, instantaneous acceleration vector samples are taken at a fixed frequency and placed into a buffer. In continuous time, acceleration is the second derivative of distance, when taken with respect to time. In the discrete domain, the riemann sum of discrete acceleration measurements can be taken up until a specific point to determine the approximate velocity at that point by the same principle. This same process can be applied a second time to the discrete buffer of velocity measurements to determine an approximate displacement. To account for the effects of noise, which are exacerbated by the process of calculating these sums, a Kalman filter will be implemented.



(https://www.math.hmc.edu/calculus/tutorials/riemann_sums/)

1. Facial feature detection portion

Video frames are captured from the back-facing camera and placed into a buffer of images. These images are processed in a callback function triggered by the arrival of new images from the video capture hardware. Buffered images are passed to a facial feature recognition algorithm, provided in a library from Apple, which determines the positions of facial features at pixel positions. Specific points are generated for the eyes and mouth, as well as height and width dimensions, for the detected face. The eye and mouth features form a triangle, whose area can be taken for each image containing a detected face. Subsequently captured facial triangles can be compared with each other to find a change in area. This change in area will be unitless, since the initial relative measurements are taken from a single vantage point.



The displacement measurement from the motion sensing module will be compared with the change in area of the facial feature triangle to find the distance the target is from the camera. When a face is first detected, this will initiate a displacement measurement. When the same face is detected from a new distance, the displacement measurement session will end and a depth will be calculated. This will continue until a face is no longer detected. Every image arrival containing a face will therefore initiate and

<u>Risk Analysis</u>

The main risk for this project

Key risks: loss of personnel

Strengths: Inexpensive Small Easy to use	Weaknesses: Attaches to device
Opportunities: Networking capabilities Potential for add-ons	Threats: More convenient independent device manufactured

Main risks:
misdiagnosis resulting from errors in the different compensation elements
software: drift in the accelerometers may cause inaccurate distance measurements to be taken

Project Management

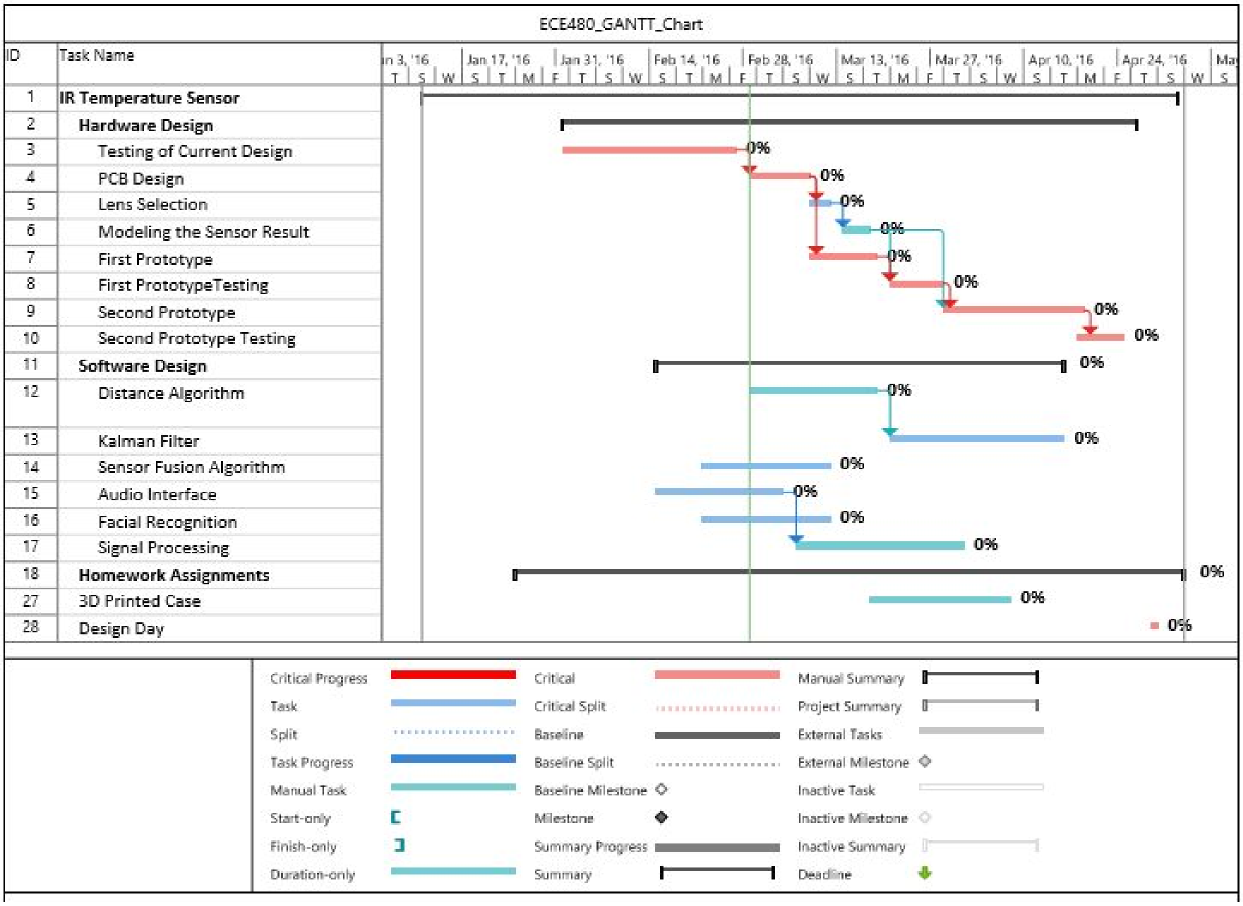
This task requires hardware and software teams working in parallel. The hardware team consists of Anish Gill, Yujie Hao, and Craig Stoddard. Ian Bacus, Yousef Gtat, and Anna Little make up the software team.

The hardware team is tasked with the PCB design and programming the microcontroller. The software team will work on programming the phone to process the temperature sensor data along with the phone's built in accelerometer, gyroscope, and camera data to display to the user an accurate result. Individual projected contributions are listed in the table below.

Ian Bacus	Anish Gill	Yousef Gtat	Yujie Hao	Anna Little	Craig Stoddard
Sensor Fusion Kalman Filter Facial Recognition	PCB Design CAD Design for 3D Case	Signal Processing Audio Interface	Microcontroller Programming Lens Selection Circuit Testing	Distance Algorithm UI Design	PCB Design Circuit Testing

Resources needed/ procurement

The resources required for this project are MacBooks and iPhones for development and testing which have already been obtained. The MSP430G2553 and the TMP007 which have already been obtained. The PCB and parts for soldering, which will be requested from the ECE shop during the seventh week of development. A phone case to hold the final prototype. This will be built on a 3D printer during the thirteenth week of development. Regarding the specific timeline for accomplishing these tasks please refer to figure the figure below.



Issues and limitations

One of the significant limitations is the PCB design and manufacture. The PCB subteam had to run two versions of THERM in parallel, one to be manufactured by the ECE shop for testing purposes and the other is for the final prototype (which will be much smaller in terms of size). This will consume a significant portion of the team's effort and time due to the limitation of manufacturing cost and resources. The manufacturing costs adds to the budget, however, the time it takes to manufacture the final circuit from a PCB company is estimated to be 2-3 weeks, thus limiting the team's ability to create multiple versions of the final product. The PCB team will merge the temperature sensor circuit interface with the hijack circuit. The thermopile range of the TMP007 is 3.5cm max range for 10cm wide face, thus requires the PCB subteam to place a lens on top of the temperature sensor to properly focus on target (i.e. patient). The size of the required lens is very small, which can cause an issue because the team speculate that the lens might have to be custom made or manufactured by third party company. The temperature sensor feeds data back to the MSP430, which is a low power IC used to communicate with the iPhone application. The MSP430 IC was specifically chosen due to some circuit power consumption constraints of a maximum of 15mW. This limits the team’s choices to use low power IC thus limiting the computational power and algorithm complexity implemented on the embedded circuit. Finally, the hijack circuit bit rate with microphone bandwidth (20 KHz) was estimated to be a maximum of 800Hz sampling frequency. As a result, the hijack circuit has constraints when it comes to real time signal processing algorithm for pattern recognition of the temperature health record.

Budget

Component	MSP430xG	TMP007	Case	PCB	Lens	Total
Price	$4.50	$4.67	$11.00	$100.00	$15.00	
Quantity	3	2	2	1	2	
Total	$13.50	$9.34`	$22.00	$100.00	$30.00	$174.84

References

B. Hijack paper
C. TMP007 Datasheet

Appendix

Application	Non-contact Temperature	Phone-based form factor	Audio Jack	Distance sensor
Temperature gun	X			
Mendipweather		x		
Yildizyan et al	x			x

HiJack		x	x	
Us	x	x	x	

Extra notes

Summary of patented device similar to ours from Yizldan

“A non-contact IR thermometer according to various embodiments of the present invention includes, among other things, an IR sensor, a distance sensor, a microprocessor, a memory configured to communicate with the microprocessor, and a user interface device configured to receive inputs from the microprocessor. The memory includes compensation information, e.g., a look-up table or mathematical equation that may be used to determine a compensated temperature of a body part based on a measurement of the same or another body part. For example, the compensation information may be used to determine a compensated temperature of a forehead based on a measured temperature of a forehead. Or, the compensation information may be used to determine a compensated oral or oral-equivalent temperature based on a measured temperature of a forehead. The IR thermometer may be configured to simultaneously or in sequence measure a temperature of the target object, the ambient temperature, or temperature of the thermometer, and a distance between the IR thermometer and the target. The microprocessor may use these values and the compensation information to determine a compensated temperature and communicate this temperature to the user interface device, which may further communicate the compensated temperature to a user. “

Email describing our project

Our project focuses on interfacing an infra-red temperature sensor with the audio jack interface of an iPhone to create a low power non-contact temperature sensor for biomedical applications. The iPhone will run an application that utilizes sensor fusion and image processing to correct for the effects of distance on the IR temperature sensor's readings. Powering of the device and communication with the iPhone app will be facilitated through the audio jack and some additional interfacing circuitry of our own design. This device is intended to aid hospital clinicians in taking temperature measurements and documenting patient information more easily.

The temperature sensor will be fastened onto the phone in a fixed position with a case, so its position relative to the camera will be fixed. Measurements from the accelerometer, magnometer, and gyroscope will be used to measure the phone's absolute orientation and relative displacement. Relative displacements will be measured against "feature shrinkage" (in the next paragraph) to determine depth. The orientation could also be used to determine the distance seen by the temperature sensor if it is necessary, although in the final design they will be placed very close together.

The compensation function will utilize ambient temperature and depth measurements to correct sampled temperature readings. The depth calculation will use the facial recognition API from Apple's Core Image framework to determine a feature triangle between the mouth and eyes of a target's face. This triangle will shrink by very small amounts when the phone is moved away from the target's face, and this can be compared with a displacement measurement (along the axis of the back facing camera, determined by the phone's orientation in space) to determine depth.

The temperature sensor will encode each temperature sample (average temperature over circular projection, onto the target, of a cone originating from the sensor) with 32 bits, and if we send this over a UART line (with extra bits for start, stop, parity on each byte) this would allow us to send somewhere around 800 samples per second at the maximum transmission rate of 20khz (2 bits per cycle) while streaming. This is a much higher rate than the frames per second recorded by the camera on the iPhone.

Regards, x