Multi-Function IoT Button

Design Document

Team 88 - Naveed Jamali, Daryl Lim, Jin Li

ECE 445 — Spring 2017

TA: Eric Clark

Table of Contents

1. Introduction

1.1 Objective

The average person’s life is far more complex than it was a hundred years ago. In this day and age, people have far more hectic schedules and must keep track of more responsibilities than they would have in the past. People are also more aware of the impact that their lifestyle has on their health. With all these new daily tasks and responsibilities, it has become harder to keep track of everything.

Our goal is to better keep track of these daily objectives by creating small, multi-purpose, IoT-enabled buttons with LCD displays. Each button can be placed in the physical location where the task or goal is to be accomplished to act as both a physical reminder and a way to see if the goal has been met. The button can be set to 3 different modes: Counter (keeps count of button presses, ex. press to record each cup of coffee to monitor caffeine intake), Checklist (timely reminder to perform a task, ex. press to show everyone in household that the dog has been fed), or scripted action (assign task to button, ex. place near bed and press to turn off all lights in home). They would connect to a device via Bluetooth to send and receive data through an iOS application, and to detect if a user is home. Our plan is to have all of features of the button to run on it’s own hardware rather than through the connected bluetooth device.

1.2 Background

Software applications already exist to aid in our remembering, including calendars and checklist applications such as Todoist [1]. However, these applications often fail to be flexible in regards to location and changing schedules, only able to send reminders at set times. To take location into account, most applications will at best react to a geofence around an entire building. However, tasks are often forgotten without a physical reminder in the exact location they are to be completed. Even if you remember or are reminded to do something in one room, upon walking into the room containing the task, you may get distracted and forget why you are there due to the “Doorway Effect” [2]. This heavily studied phenomenon explains the common memory loss that occurs upon crossing physical barriers, such as when entering another room or getting into a car.

Our button would be an elegant solution to this problem. By putting a button right where the task is to be completed, the button will beep with increasing frequency and intensity as the due time approaches to draw the user into the room to complete the task. This would prevent the doorway effect mentioned above and allow some flexibility in when tasks need to be accomplished. Users could view the data regarding their daily habits collected from the buttons on their smartphones through a bluetooth connection. The buttons could also carry out actions such as sending texts or controlling lights through the smartphone, making them IoT connected. As the inspiration for this project came from the Amazon IoT (Dash) button, we plan for our button to have a similar form factor. The button should also be similarly priced (although the display, microphone, and speaker will add significant cost) and have a comparable battery life.

1.3 High-Level Requirements

- Buttons must be able to connect to an application on the phone or computer wirelessly (through Bluetooth) to allow data collected to be stored on it.
- Buttons must be able to recognize their currently desired function and correctly carry it out independant of all other features (including the connection to the smartphone).
- Buttons must consume as little power as possible to prolong battery life, ideally able to last weeks to months.

2. Design

2.1 Block Diagram

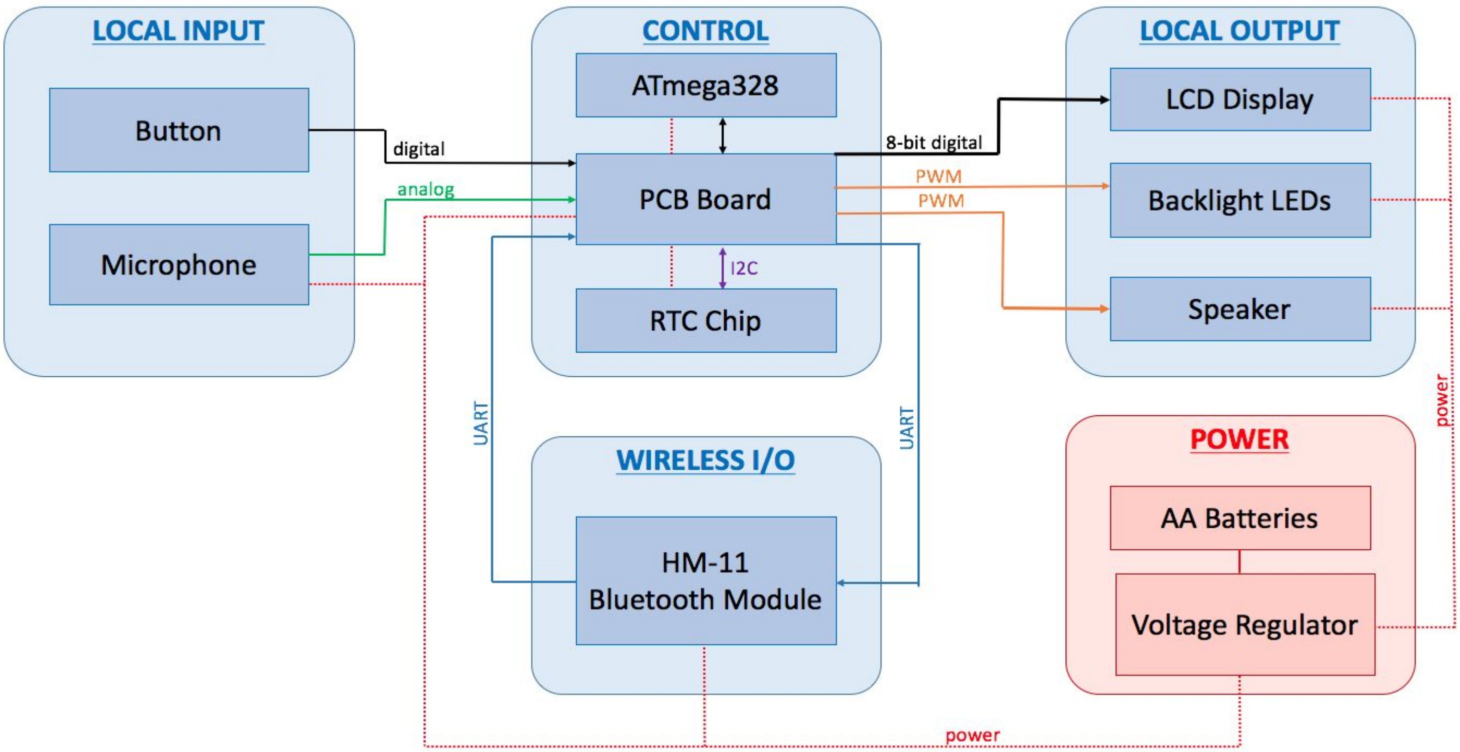

Figure 1: Block Diagram

Our project can be divided into five parts: Power, Local Inputs, Local Outputs, Control, and Wireless IO. The power supply (battery-powered) powers the button continuously regardless of time of day and should be able to maintain power for weeks to months before requiring changing. The local inputs send the designated data to control to properly handle it. The control then decides the reaction and sends the correct response to local outputs. Our wireless IO of choice (Bluetooth) connects our system to the user device to send and receive data. It allows our buttons to be initialized by an iPhone and detects if the user is away to enter a power-saving mode.

2.2 Circuit Schematic

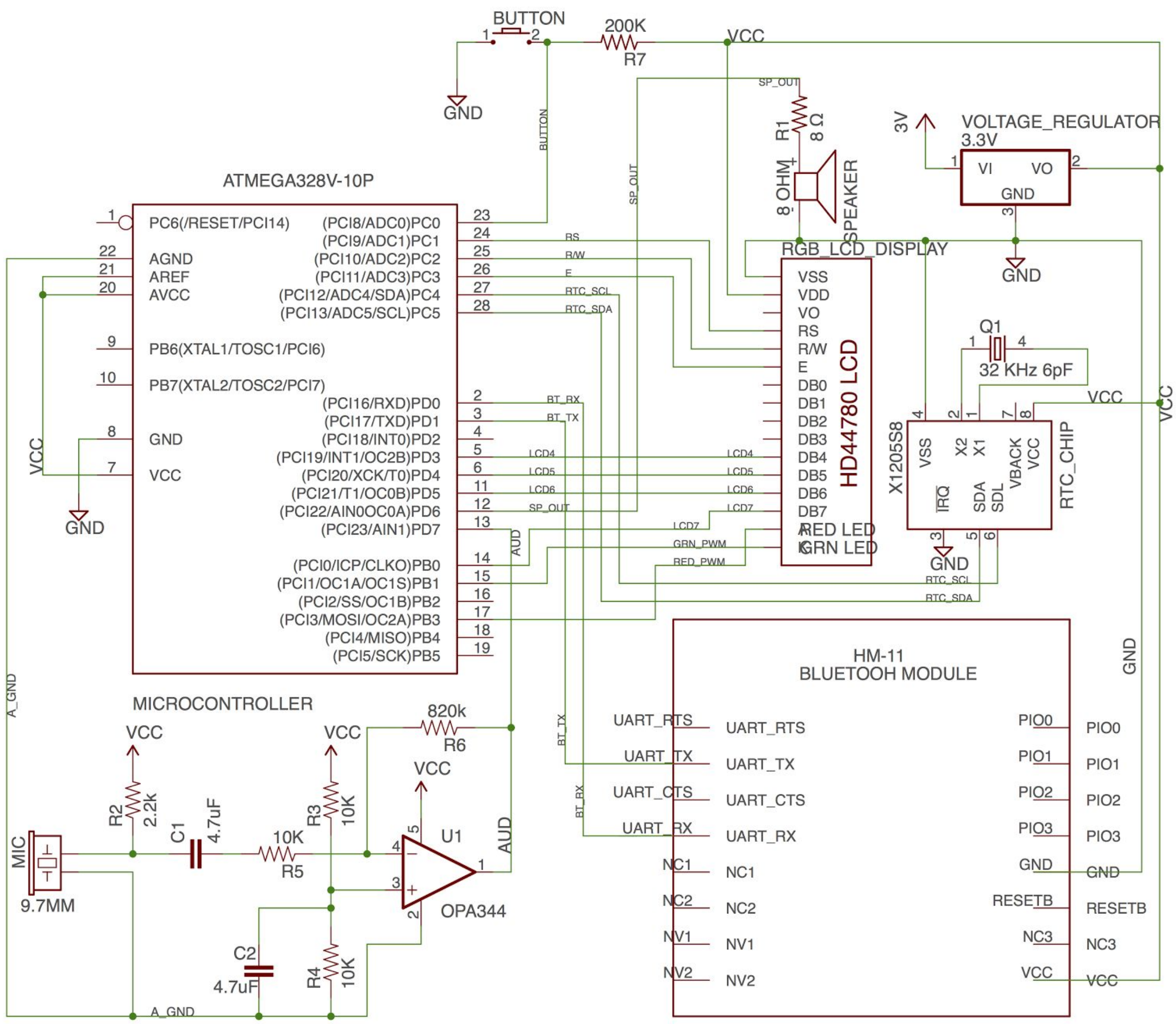

Figure 2: Circuit Schematic

2.3 Block Design

2.3.1 Power Supply

A power supply is required for each individual button to keep them functioning as long as the user is present. For our buttons, basic batteries should suffice to power it in its entirety.

One of the key intended features of our button is to be able to last for weeks to months before a battery change, as frequent battery changes on buttons would prove tedious.

2.3.1.1 Batteries

Our choice for batteries is a pack of four Alkaline AA battery [3], two connected in series, in parallel to two connected in series, which gives a combined output voltage of 3 V, and a capacity of about 2779 x 2 = 5558 mAh [4].

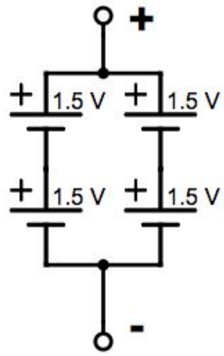

Figure 3: Battery Connection

The ATmega328P runs at 3.3V, 3.58 mA when active, and at 3.3V, 0.0045 mA when powered down. [5] Assuming the most power consuming scenario of the ATmega328P being active around the clock, we have power consumption of:

$$3.58\text{mA} \times 24\text{h/day} = 85.92 \text{ mAh/day}$$

For the RTC chip, the maximum current "IMAX would therefore be calculated as follows: IMAX = (3.3V - diode drop) / R2 ≈ (3.3V - 0.7V) / 2kI ≈ 1.3 mA".

In a day, the power consumption of the RTC chip should be:

$$1.3 \text{ mA} \times 24\text{h/day} = 31.2 \text{ mAh/day}$$

For the LCD display, when it works as LED, a single color draws 20 mA.

Assuming the LED is on for 20 seconds a day, that'd be:

$$20 \text{ mA} \times 20/3600 \text{ h/day} = 0.1111 \text{ mAh/day}$$

For our HM-11 chip (Bluetooth), we created an optimized sleep/wake cycle to have the chip only be set active for as little time as possible and set to deep sleep for the remainder in order to conserve battery life. All of the calculations for this are found in section 2.3.2. We found the average current draw of the HM-11 with these constraints to be 0.608mA. So the average power consumption will be:

$$0.608 \text{ mA} \times 24\text{h/day} = 14.6 \text{ mAh/day}$$

The speaker has an impedance of 8 Ohms, and a power consumption of 0.3 W, so the current draw would be 0.194 Amps whenever the speaker is making sounds.

Assuming the speaker makes sounds for 20 seconds a day,

$$194 \text{ mA} \times 20/3600\text{h/day} \approx 1.08 \text{ mAh/day}$$

The voltage regulator's quiescent current is negligible, as it is <<1 mA for input voltages close to 3.3V.

With our initial battery capacity of 5558 mAh, we find that the number of days our batteries should be able to supply power to our design should be:

$$5558 \text{ mAh} / (85.92+31.2+14.6+1.08) \text{ mAh/day} = 41.9 \text{ days}$$

Not only does this fit our requirement of the battery life lasting from weeks to months, but whenever the user leaves the house or commands the button to be inactive, the ATmega328p would power down, shifting to using 0.0045 mA instead. Given a scenario of a single individual working a standard 8-hour work day, the ATmega328p would power down during that time, giving us a power consumption instead of:

$$0.0045 \text{ mA} \times 8\text{h/day} = 0.036 \text{ mAh/day}$$

$$3.58\text{mA} \times 16\text{h/day} = 57.28 \text{ mAh/day}$$

Or a total of 57.28 + 0.036 mAh/day = 57.316 mAh/day, or a battery lifespan of:

$$5558 \text{ mAh} / (57.316 +31.2+14.6+1.08) \text{ mAh/day} \approx 53.3 \text{ days}$$

2.3.1.2 Voltage Regulator

The voltage regulator must provide a steady supply of constant voltage to all components. It will step up the voltage from 3V to 3.3V, as all components in our design operate at 3.3V input.

In Figure 4below, we can see that Pololu regulator we chose has good efficiency of (power out)/(power in). It can be observed from the three different curves that the efficiency increases as the input voltage increases. At around 100 mA, when the input voltage is 2.4V, the efficiency is around 85%. With our input voltage of 3V, we expect the efficiency to be even higher.

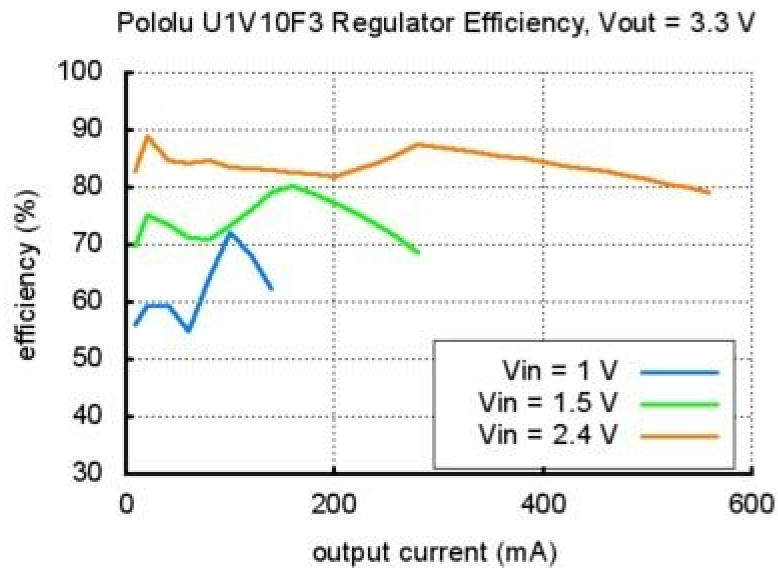

Figure 4: Regulator Efficiency vs Output Current

Referenced from "Pololu 3.3V Step-Up Voltage Regulator U1V10F3" [6]

Requirement	*Verification*
Output voltage from the voltage regulator must be 3.3 ±0.3 V, while providing a current ranging from the lowest possible (2 mA) to highest possible (233 mA) current draw of our design.	1. *Set the input to the voltage regulator to be 3.0V (2 AAs in series) using a DC power supply* 2. *Test output voltage across both 1650 Ω (simulating lowest power scenario) and 15 Ω (simulating highest power scenario) resistor load.* 3. *Ensure that the voltage is within the range of 3.0 - 3.6 V.*
Output voltage from the voltage regulator must be 3.3 ±0.3 V given a input voltage ranging from 3.0V to 1.5 V (to simulate the voltage drop of 2 AA batteries in series)	1. *Repeating the steps above, slowly dial the DC power's output voltage from 3.0 down to 1.5 and verify that the output voltage of the voltage regulator remains within the range of 3.0 - 3.6 V.*

Table 1: Requirement and Verification for the the power module

2.3.2 Wireless IO

2.3.2.1 Bluetooth

The purpose of the Bluetooth module is to:

1. Transmit and Receive data from a host device
    a. Receive programming instructions to set time, mode, label, etc.
    b. Transmit data from button presses to be recorded in host device application
2. Detect Presence of User
    a. If user is out of range, then enter low-power mode. (LEDs off, Speaker disabled, machine learning voice recognition disabled).

The HM-11 connects to the ATmega328 using the UART protocol. The SPP profile would be most appropriate for our purposes. The input voltage to the chip is the same as the rest of our circuit (3.3V).

The data sheet on the chip states that its maximum range is 30 meters in an open space. For our purposes, we require that the button recognize the user within 20 meters in an open space and 8 meters going through 2 walls. This is reasonable because, as stated above, one of the main purposes of our Bluetooth module is to enter a low power mode if the user is to far away to hear the speaker or issue commands to the microphone. 20 meters is a fairly large distance away from the button where the user probably would not hear it nor want to interact with it. It also logically follows that if walls attenuate sound and decrease audible range, it would be alright if the bluetooth range decreases as well. Therefore, we set the range requirement given two average apartment walls (drywall, not concrete) to be 8 meters, which is the distance from the living room to two rooms down in a typical apartment.

In order to conserve power, we will only activate the bluetooth chip for only a fraction of the the time and leave it in a low-power sleep mode as long as possible. However, we want to ensure that if a user were to walk to the button from outside its range, press it, then immediately walk back out of range that the button will have enough time to react and sync data at least once while both approaching and leaving the button. Given the worst case range of 8 meters and the average human walking pace of 1.4 m/s, this implies that the **Bluetooth chip needs to be activated at least once every 5.7 seconds** to meet this requirement.

As for how long the chip needs to be active, the HM-11 advertises a data transfer rate of 115,200 bits/sec. Once data transfer cycle would consist of a send and receive portion, which be estimated as follows:

Sent to button:

Program Enable (1 bit)	Mode (2 bits)	Label (192 bits)	Mode-Specific Data (55 bits)

Program Enable:	True when programming button. Rest of bits are don't cares when false
Mode:	00 - Set Clock 01- Counter 10 - Checklist 11 - Action
Label:	Displayed on LCD, 24 chars x 8 bits per char = 192 bits
Mode - Specific Data:	Worst case is for the checklist. There are 1,440 mins in a day, which requires 11 bits to represent in binary. This means each alarm requires 11 bits, and if we allow up to 5 alarms a day we need 55 bits

Total: 250 Bits

Sent by Button:

The data sent by the button is entirely dependant on the mode it is set to. For the counter, the button will probably not be pressed more than 128 times (probably far less) a day, so 8 bits the hold the count should suffice. The checklist only needs to hold a boolean value for every alarm to represent whether or not the task has been completed. With 5 daily alarms, 5 bits are needed. For the scripted action, we may allow 3 different actions to be carried out depending on how the button is presses. For example, if the button is used to control lights, pressing it briefly can turn the lights on/off, pressing and holding it can dim them, and double pressing it in quick succession can reset the brightness to maximum. We would need 2 bits to represent these 3 different types of button presses. Therefore, the worst case for the number of bits to be sent by the button is for the counter at 8 bits.

Putting these two components together, that means that the minimum amount of data that needs to be sent/received per data transfer cycle is 258 bits. In order to allow large tolerance for error and to account for extra parity bits and bits required to control the Bluetooth protocol, let's call this 512 bits.

The HM-11 claims to have a data transfer rate of 115,200 bits/sec. With the 512 bits we wish to be able to transfer per cycle, that means that the **chip must be on for 4.44 ms to complete one data transfer cycle.**

Combining our constraints for how often the chip must be be activated with how long it needs to be on, and adding even more of a tolerance to each constraint, it should suffice to **have the chip be set active for 5 ms out of every 5 sec**, and set to deep sleep for the remaining time.

The average current draw with this specification is calculated using the following table and equation:

Mode	Current (mA)
Transmit	15
Receive	8.5
Deep Sleep	0.6

Table 2: Current Drawn in Different Modes (datasheet)

$$i_{avg} = \frac{((.005\ sec)\ ((\frac{250\ bits}{258\ bits})(8.5\ mA_{\ receive}) + (\frac{8\ bits}{258\ bits})(15\ mA_{\ transmit}))_{\ active} + (4.995\ sec)(0.6\ mA_{\ deep\ sleep})}{5\ sec}$$

Equation 1: Average Current Draw of HM-11

Solving the equation above, we get an **average current draw of 0.608 mA** for the HM-11.

Requirement	*Verification*
The Bluetooth chip must be able to detect if the user device is within 20 meters in open space, and within 8 meters if there are two walls in between the button and the user device.	*1. Hold the user device in one hand and measure the distance between the person and the button, make sure it is 20 meters in open space.* *2. Pair user device (phone, laptop, or tablet) to the button, verify connection in the user device's built in settings* *3. Hold the user device in one hand and stand 8 meters away from the button, placing 2 non-concrete or stone walls between the phone and the button .* *4. Pair user device (phone, laptop, or tablet) to the button, verify connection in the user device's built in settings*

The Bluetooth chip must be able to detect a user's presence within 5 seconds.	*1. Set the HM-11 to active and pair it with a host device.* *2. Put the MH-11 to deep sleep and prepare a stopwatch.* *3. Set the HM-11 back to active and time how long it takes to reconnect.*

Table 3: Requirement and Verification for the Bluetooth Module

2.3.3 Local Inputs

2.3.3.1 Button

We intend to adapt from the design of the Amazon dash button. It should register the press from a user, and update relevant information (ex. that the user has taken his/her daily medicine). The button itself would not have an attached debouncing circuit; we intend to debounce the button input in software using the following logic [7].

```
void loop() {
  // read the state of the switch into a local variable:
  int reading = digitalRead(buttonPin);

  // check to see if you just pressed the button
  // (i.e. the input went from LOW to HIGH),  and you've waited
  // long enough since the last press to ignore any noise:

  // If the switch changed, due to noise or pressing:
  if (reading != lastButtonState) {
    // reset the debouncing timer
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {
    // whatever the reading is at, it's been there for longer
    // than the debounce delay, so take it as the actual current state:

    // if the button state has changed:
    if (reading != buttonState) {
      buttonState = reading;

      // only toggle the LED if the new button state is HIGH
      if (buttonState == HIGH) {
        ledState = !ledState;
      }
    }
  }
```

Figure 5: Code in Arduino for Button Debouncing [7]

2.3.3.2 Microphone

The purpose of the microphone is to capture voice commands and transfer analog signal to the ATmega328p chip, where the analog signal will be converted to digital, and processed for word recognition. The microphone can pick up sounds within the frequency range of 100 ~ 10,000 Hz [8], while for human speech, “Normal voice range is about 500 Hz to 2,000 Hz” [9], so the speech frequency should be covered by the frequency range the microphone can pick up.

Requirement	*Verification*
The microphone’s maximum output voltage is amplified to 1 to 5 Volts.	*1. Set microphone to receive audio input.* *2. Using a computer or tone generator, set frequency sweep tone from 500 to 2000 Hz, at 70 dB very close to the microphone (within 50 cm)* *3. Use an oscilloscope to measure the voltage after the amplifier, to make sure that the audible part of the signal has peaks in the 1-5 Volt range and no peaks are higher than 5 Volts.*
The microphone must be able to accurately detect speech within a 10 foot radius.	*4. Set microphone to receive audio input.* *5. Stand at or within 10 feet from the mic and make audible sound, checking to see if it’s detected.* *6. Directly read the voltage across the amplifier, make sure voltage is between 1-5V.*

Table 4: Requirement and Verification for the Microphone

As the microphone's natural voltage is small (found to be in mV) and can't be read, thus in our design we need an amplifier to amplify it to a discernible range. Ideally we want 5V however, the voltage should enter a discernible range at 1V. So we intend to amplify the voltage to be at least 1V. Since the sensitivity of the microphone is at -46 dB, we will need an amplifier of 46 dB to amplify the output voltage to 1V, assuming that the mic will be receiving 94 dB-SPL signals. However, we must recognize that the input intensity to the mic will not always be as high as 94 dB-SPL when capturing speech. When a person is shouting at 1 ft away from the microphone, the voice level is 88 dB-SPL; when a person is 12 ft away from the microphone talking at a normal volume, the voice level is 48 dB-SPL [10]. Assuming the voice level can be as low as 48 dB-SPL, we will need an additional 94-48=46 dB. Therefore, we need a total of 46 dB (from voice) + 46 dB (from speaker) = 92 dB for our amplification of sound.

2.3.4 Local Outputs

2.3.4.1 LCD Display

An LCD display that must be able to display various pre-programmed instructions ("take medicine"), feedback ("task accomplished, good job!"), etc. corresponding to its current functionality.

Requirement	*Verification*
The LCD display must be able to display the right message when prompted.	1. *Implement simple program to output desired phrase (ex. Hello World!)* 2. *Observe display for desired phrase.* 3. *Test three times with three modes.* 4. *Observe display: should display task on button press.*

Table 5: Requirement and Verification for the LCD Display

Pin	Symbol	Description
1	Vss	GND
2	Vdd	Power Supply
3	E	Enable
4	DB0 - DB7	Data Bus Software
5	LED +	LED unit anode
6	LED -	LED unit cathode

Table 6: LCD Display Key Pin Configurations

2.3.4.2 LED

A visual color display that must be able to react when time indicated for task completion approaches. Since our LCD display has three colors available, we will utilize two of those colors to act as our "Encouraging" and "Discouraging" colors. Blinks on and off until button is pressed or command is given. The display takes a forward voltage of 3.3 V and the changing of colors can be taken care of by the PWM.

Requirement	*Verification*
LCD display must blink green when Counter is < 3 steps (at 2 from and 1 from) from the "goal" and blink red when Counter is < 3 steps from the "limit"	*1. Simulate almost reaching goal/limit by pressing the Counter up to 3 before goal/limit.* *2. Press button once. Verify that the button blinks the appropriate color (red for "limit", green for "goal).* *3. Repeat and verify that it blinks the appropriate color again.* *4. Repeat steps 1-3 for the other color.*

Table 7: Requirement and Verification for the "LED"s (LCD Color Display)

2.3.4.3 Speaker

A speaker that must be able to react when time indicated for task completion approaches. Beeps until button is pressed or command is given.

Requirement	*Verification*
The speaker must be able to beep with a frequency of 1600 Hz ±50 Hz to alert users of approaching time limit, with an "Encouraging" frequency of 600-800 Hz, and a "Discouraging" frequency of 200-400 Hz.	1. *Program speaker to beep at designated frequency.* 2. *Record sound and display the spectrogram of sound in Matlab to identify frequency component.* 3. *Verify frequency output is within desired range (1600 Hz* 4. *±50Hz for general alert, 600-800 Hz for "Encouraging", and 200-400 Hz for "Discouraging").* 5. *Repeat steps 1-3 both for "Encouraging" and "Discouraging" frequency ranges.*
The speaker must be able to beep with intensity > 60 dB and < 85 dB.	1. *Stand within 20 meters (if in open space) or 8 meters (if two walls in between) of the speaker.* 2. *Utilize sound-level meter to measure decibel level of speaker sound.* 3. *Verify that decibel level is between 60 dB and 85 dB.*

Table 8: Requirement and Verification for the Speaker

For intensity ranges, we chose a decibel level that surpassed that of human speech (60 dB) to be able to make enough of an impression to spur users into action and lower than a level that could possibly damage hearing after long exposure (85 dB) to be considered within a safe range. As an alarm clock usually functions at 80 dB we consider this would be at an appropriate level.

2.3.5 Control

2.3.5.1 Microcontroller

In the control unit, we will use a microcontroller using AutoDesk EAGLE, which will comprise of a Real-Time Clock (RTC) timekeeping chip, a ATmega328p processing chip, and circuit elements designed by us.

In the control unit, we will implement one large state machine used in the operation of the button. The state machine will have three main branches for each mode the button is set to. State transitions will react to button inputs as well as the output of the serial Real-Time Clock (RTC) timekeeping chip to be mounted on the PCB.

The microcontroller should be powered by our batteries with the voltage stabilized by the voltage regulator, and should be the center of connection for all functional components in our design: the microphone input, button input, speaker output, LCD display, LED display, and bluetooth connection.

We intend to mount an Arduino chip (ATmega 328p) onto our PCB, in order to utilize its processing powers to run word recognition algorithms on extracted features of the audio input. Our previously considered approach is to pre-train the weights and biases of a fully-connected neural network on a laptop for specific words, and deploy those weights and biases onto Arduino for real-time recognition. Our alternative approaches can be dynamic time warping, a CNN, HMM (Hidden Markov Model), or even a simple K-Nearest-Neighbor classifier with a pre-recorded database for feature matching.

Below are the requirements and verifications of the control module in our block diagram design.

In exploring different methods for word recognition, we compared four different kinds of classifiers for a fully-connected neural network.

Below is a convergence plot for four different kinds of classifiers, on a dataset of consonants from an experiment by Bowon Lee et al., 2004 [11]. As can be seen, a logistic classifier or a SVM classifier is more ideal than a linear classifier or a simple perceptron classifier in terms of how quickly the classification error plot converges.

However, it is worth noting that the four learning rates used for gradient descent are different from each other. This is because if we use the same learning rate, it might be too big for some classifiers, and the gradient "steps over" the error metric's local minima. If a chosen learning rate is too small, then convergence is slow; however if a chosen learning rate is too large, then the error rate might not converge at all due to the large steps taken [12]. With that trade-off, we chose learning rates for each classifier accordingly.

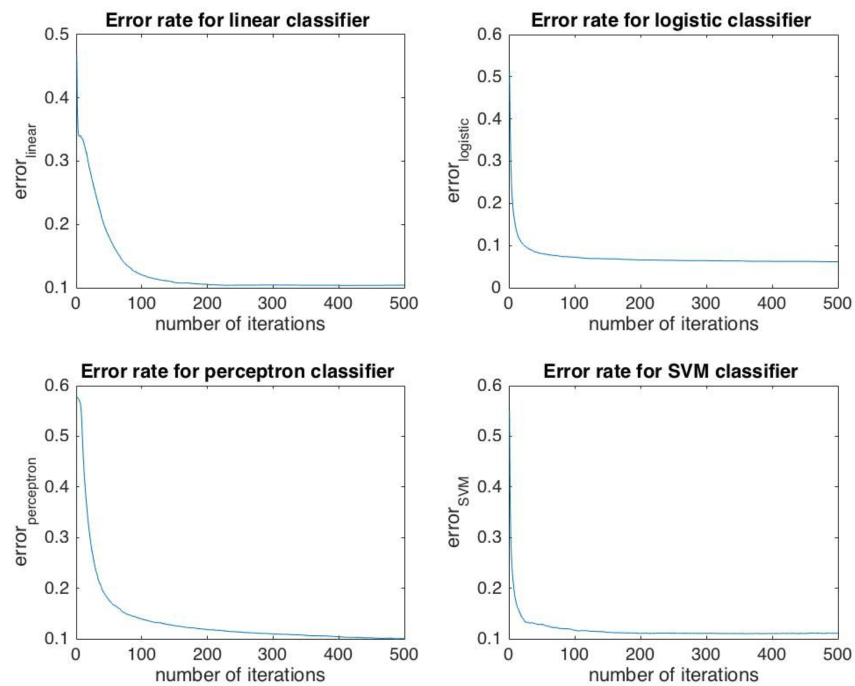

Figure 6: Convergence Plot for Four Classifiers
**error denotes the classification error percentage of the testing data*

The following table shows the error metric (different from the classification error rate) of the four different classifiers we explored, and their corresponding gradient equation for gradient descent.

	Error Metric	Gradient Equation	Fixed Learning Rate

linear	$E = \sum_i ((t_i - y_i)^2) = \sum_i ((t_i - g(w'x_i + b))^2)$ $g(a) = a$	$\frac{dE}{dW_j} = -2 * \sum_i (t_i - (w'x_i + b)) * x_{ji}$	0.00001
logistic	$E = \sum_i ((t_i - y_i)^2) = \sum_i ((t_i - g(w'x_i + b))^2)$ $g(a) = \frac{1}{1+exp(-a)}$	$\frac{dE}{dW_j} =$ $-2 * \sum_i (t_i - y_i) * y_i * (1 - y_i) * x_{ji}$	0.0001
Perceptron	$E = \sum_i (max(0, -(w'x_i + b) \cdot t_i))$	$\frac{dE}{dW_j} = \sum_{errors} (-x_i \cdot t_i)$ [13]	0.000005
SVM	$E = C * (\|w\|^2 + \sum_i max\ (0,\ 1 - t_i * (w'x_i + b)))$	$\frac{dE}{dW_j} = C * (2w_j + \sum_{errors} (-x_i \cdot t_i))$ [13]	0.0001

Table 9: Theoretical Basis for Four Classifiers Explored
** t denotes true labels, y denotes predicted labels, g conventionally denotes a non-linearity, w denotes weights, b denotes biases, x denotes input data, and C is a tunable constant.*
*** Code for carrying out the above procedures and calculations adapted from team member Jin Li's work in ECE 544 Fall 2016.*

The timing requirements for our inputs/outputs are purposefully lenient since humans react in orders of magnitude slower than a microprocessor's clock cycle, and for the most part we our operations are not computationally intensive.

Requirement	*Verification*
Microcontroller must be able to correctly read button inputs and set proper outputs within 5ms	1. *Connect the button input of the ATmega328 to a square wave generator set to output 0-3.3V, 50 Hz, at a 50% duty cycle. This simulates the button being pressed for 5ms then released for 5ms.* 2. *Program the ATmega328 to set the output of one of its pins to match the button input* 3. *Connect oscilloscope probes to both the input and output pins* 4. *Turn on the waveform*

	generator 5. Observe the two waveforms. Ensure that the delay is less than 5ms (output pin is set to high before the input goes low).
Word recognition rate must be > 60%.	1. *Each of the three teammates will speak into the microphone, from distances ranging from 0 ft to 10 ft, for a total of 10 utterances.* 2. *Must correctly recognize command and stop at least 18 of the 30 utterances.*

Table 10: Requirement and Verification for the Control Board

2.3.5.2 RTC Chip

The Real-Time Clock (RTC) we chose is the DS1339A chip by Maxim Integrated. The purpose of incorporating this clock chip is to keep track of the time in a day for setting reminders, having a time span reference when counting, etc.

Requirement	*Verification*
The RTC must be able to correctly recognize time set by device within 30 seconds.	1. *Connect RTC chip to Atmega328 and connect to Arduino IDE on computer.* 2. *Run 'date' to check time set on RTC chip.* 3. *If time is not accurate, connect to Internet to sync clock time to Internet time.* 4. *Repeat steps 2-3 until time set on RTC is accurate.*

Table 11: Requirement and Verification for the RTC Chip

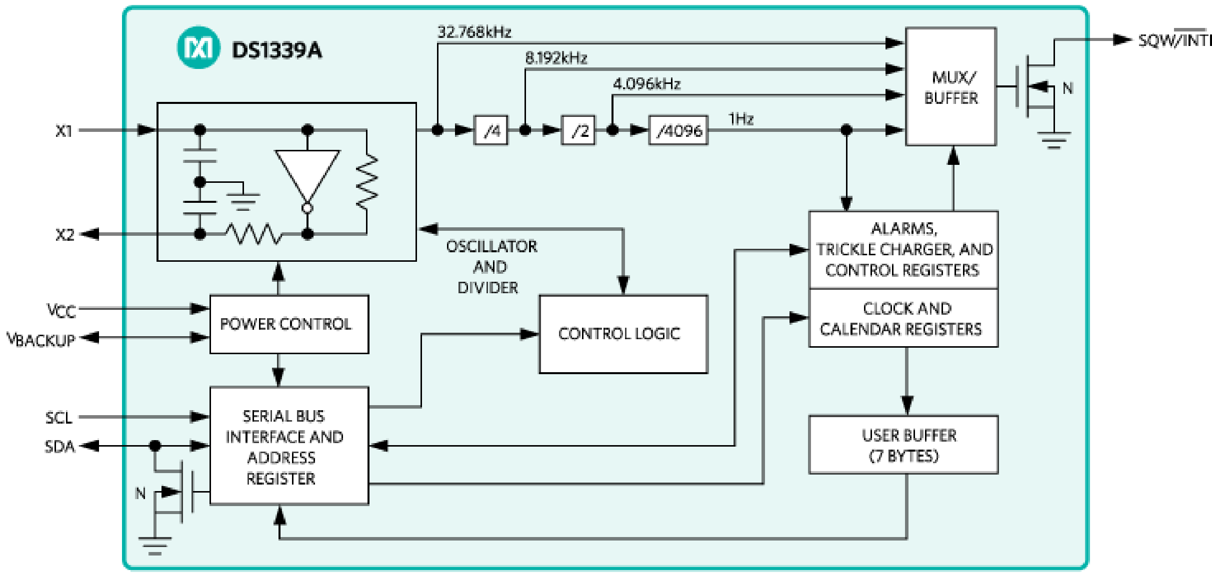

Figure 7: RTC Chip Block Diagram [14]

This chip will communicate with the rest of the PCB board via I2C serial interface; it takes an input voltage from 1.8 to 5 V.

The RTC chip has an internal oscillatory circuit with a preset load capacitance of 6 pF and is designed to take a standard 32.768kHz Quartz crystal.

2.4 Tolerance Analysis

In our design of the microphone amplifier, we intend to adapt from the amplifier circuit in [15]. In the graph below, we can see the "same gain across the frequency spectrum that is picked up by the mic" [15] for actual and amplified signals, in the solid black and solid burgundy curves. Furthermore, because human speech is only from 500 - 2000 Hz, the two responses should be entirely identical, by looking at Figure 8 below. From the website, we gather that "The input of the amplifier is biased at ½ VCC... The output ... is also at ½ the supply voltage, so it can be connected directly to the ADC of microcontroller", so the output should be at ½ x 3.3V = 1.15 V.

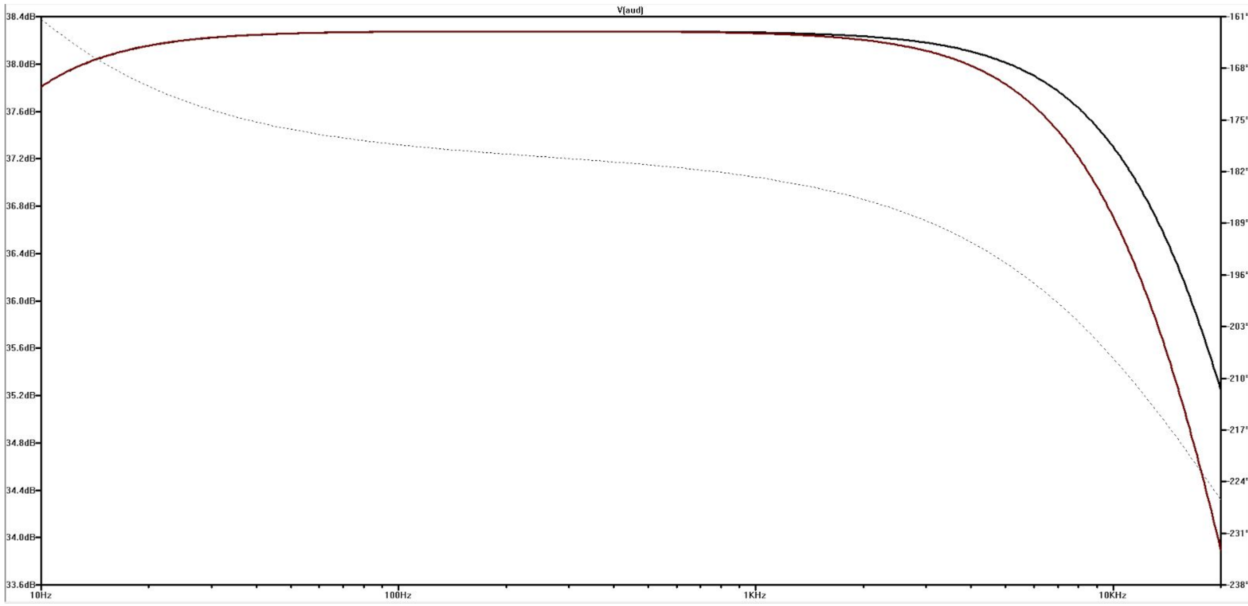

Figure 8: Frequency Response of the Microphone Amplifier Circuit [15]

(Burgundy Curve - Amplified Audio Signal;

Black Curve - Original Audio Signal;

Dotted Black Curve - very small AC voltage output)

2.4.1 Microphone Sensitivity Deviation

According to the datasheet of the microphone [8], there could be a deviation of 2 dB from it's -46 dB sensitivity. If the deviation is in the range of [0, 2] dB, then the sensitivity would be in the range of [-46, -44] dB, which will not cause any problems, because we precalculated 46 dB for the pure microphone amplification (as well as another 46 dB making up for sound intensity less than 94 dB-SPL). If the deviation is in the range of [-2,0] dB, then the sensitivity would be in the range of [-48, -46]. We could still amplify a sound signal to full scale with the 92 dB amplifier if the speech intensity is larger than 50 dB: 48 dB of amplification would counteract the sensitivity, we have 44 dB remaining for making up for the small intensity of sound, and 44 + 50 = 94 dB, which is what we assume to be the input intensity to the microphone.

In conclusion, with a [-2, 2] deviation in microphone sensitivity, the minimum sound intensity that can be picked up by the microphone can rise from 48 dB to 50 dB; the 2 dB difference is 1.258 in ratio, meaning the sound intensity will need to be 25.8% higher.

Analysis: At the worst case scenario where the sensitivity deviates to -2 -46 dB and the required minimum sound intensity should be 50 dB, our design still wouldn't completely break down. The reason is that, when using the speech recognition feature, if the users say "stop" once and the reminder sound does not stop, it is natural for us to raise our volume and keep saying "stop" until the reminder stops.

2.4.2 Amplifier Gain Deviation

The gain of the amplifier is set by the ratio of two resistors, R5 and R4 in the schematic, as well as the R_L values in the OpAmp [16]. In the ECE shop, it is common to find 1% tolerance resistors of all resistances, so the deviation of ratio of two resistors can take values in [0.99/1.01, 1.01/0.99]=[0.980, 1.020] = [-0.175 dB, 0.172dB], and we have a amplification deviation of 2%.

Analysis: Since [-0.175 dB, 0.172dB] is a trivial deviation compared to the 92 dB amplification, we can tolerant the 1% resistance deviation for resistors.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

If we assume a reasonable salary to be $30/hour, which is common for internships in the Bay Area, then the total cost for developing our design project should be:

$30 /hour x 3 (team members) x 2.5 x 8 (hours/member/week) x 13 (weeks)

= $23,400

3.1.2 Parts

Description	Manufacturer	Part#	Quantity	Cost
Physical Button + shell	3D Printing it ourselves	N/A	1	$ 2
Push Button Switch	Corporate Computer	Corpco-x-6*6*6mmTacSw-10Pk	1	$0.529
Microphone	Challenge Electronics	CEM-C9745JAD462P2.54R	1	$ 5.95
Speaker	CUI	CDMG15008-03A	1	$ 2.58
Bluetooth HM11	Board: Chip: Texas Instrument	SKU 317030001	1	$ 4.72
ATmega328p Chip	ATMEL	ATMEGA328P-PU	1	$ 5.95
Voltage Regulator	Pololu	U1V10F3	1	$ 4.49
AA Batteries	Energizer	N/A	4	$ 1.7
LCD Display with RBG Backlight	adafruit	398	1	$12.95
RTC Chip	Maxim	DS1339A	1	$1.60
Crystal Oscillator	MultiComp	56P2811	1	$0.72

Table 12: All Parts Needed for the Project

Parts Cost = $43.19

3.1.3 Grand Total

Grand total = Labor + Parts = 23,400 + 43.19 = $23,443.19

3.2 Schedule

Week	Jin	Daryl	Naveed
2/20/2017	Work out the details of the design document: focus word recognition/tolerance analysis	Work out the details of the design document: focus power, input/output	Work out the details of the design document: focus block diagram/circuit
2/27/2017	- Design Review - Order parts: microphone, speaker, bluetooth hm11, ATmega328p; participate in the button design for 3D printing	-Design Review - Order parts: voltage regulator, AA batteries, LCD display, RTC chip; participate in the button design for 3D printing	-Design Review - Design button for 3D printing
3/6/2017	-Design Document final fixes: tolerance analysis, voice recognition section -Connect microphone to the ADC module on the ATmega328p chip; connect speaker into the circuit; start writing code for word recognition	-Design Document final fixes: block requirements and verifications, safety statement -Work on schematic + PCB design in EAGLE	- Design Document final fixes: Bluetooth section, schematic -Work on schematic + PCB design in EAGLE
3/13/2017	Further develop code for word recognition; test speaker	Finalize PCB design and submit request to ECE shop	Print button using 3D printing; participate in schematic + PCB

	characteristics		design
3/20/2017	Spring Break	Spring Break	Spring Break
3/27/2017	-Test the algorithmic operations on the ATmega328p chip	-Work on PCB -Test power requirements are met	-Work on PCB -Test Bluetooth requirements are met
4/3/2017	-Finish all code on word recognition -Make sure all connections related to word recognition is functional	-Finish all installations -Check Bluetooth + PCB connections	-Finish all installations -Check PCB connections to all other parts
4/10/2017	-Test and debug whole system -Test speaker/mic requirements are met	-Test and debug whole system -Test RTC functionality -Test for safety	-Test and debug system -Test controller requirements are met
4/17/2017	- Work on individual final paper (Jin) - Mock demo	- Work on individual final paper (Daryl) - Mock demo	- Work on individual final paper (Naveed) - Mock demo
4/24/2017	- Work on individual final paper (Jin) - Fix any outstanding issues from mock for presentation - Present projects	- Work on individual final paper (Daryl) - Fix any outstanding issues from mock for presentation - Present project	- Work on individual final paper (Naveed) - Fix any outstanding issues from mock for presentation - Present project
5/1/2017	- Turn in lab notebooks - Return lab supplies	- Turn in lab notebooks - Return lab supplies	- Turn in lab notebooks - Return lab supplies

Table 13: Schedule

4. Ethics and Safety

We made several safety decisions throughout the design phase of this project.

The first is in regards to our battery choices for the power module. We highly valued long battery life for our button for the sake of convenience for the user and possibly safety concerns with buttons running out of battery while being used for important tasks such as keeping track of medicine. With this is mind, high performance batteries like lithium ion would be the most useful, however we ultimately decided on alkaline batteries as a power supply to lower alternative risks in regards to leakage and other safety concerns. Additionally, not only are alkaline batteries relatively harmless in physical health, they are more convenient to acquire as well, again benefitting convenience for the user.

Battery choice aside, the main points of concern would have to be more ethically related than physically - concerns such as users being dangerously reliant on the buttons, such as relying on solely the button to keep track of intake of medicine in which case malfunctions could be deadly. In accordance with the IEEE Code of Ethics, #1: “to accept responsibility in making decisions consistent with the safety, health, and welfare of the public” [17]. Also, if the use of an emergency help application were to malfunction and the user believes help is on the way when it isn’t, this would prove to be fatal as well. As stated in the ACM Code of Ethics and Professional Conduct #1.2: “One way to avoid unintentional harm is to carefully consider potential impacts on all those affected by decisions made during design and implementation.” [18]

Solutions to this issue, on top of making sure the button works as intended as often as possible through heavy testing, is to include cautions to not be overly reliant on the buttons in case malfunctions occur. Additionally, in the case of the emergency help application, we could include a message on the LCD screen assuring that the message ad been sent. Also in accordance to ACM Code of Ethics and Professional Conduct #1.2, we decided to use an existing Bluetooth model in place of creating our own as our experience in this particular area of expertise was lacking, thus avoiding potentially harmful effects of deciding to design one ourselves.

Citations

[1] Reminders - Todoist. [Online]. Available: https://support.todoist.com/hc/en-us/articles/205348301-Reminders. [Accessed: 06- Feb- 2017].

[2] Z. Lawrence and D. Peterson, "Mentally walking through doorways causes forgetting: The location updating effect and imagination," Nov. 20, 2014. [Online]. Available: http://www.tandfonline.com/. [Accessed: 10-Feb- 2017].

[3] “Energizer Max Alkaline AA Battery”, 2017. [Online]. Available: http://www.homedepot.com/p/Energizer-Max-Alkaline-AA-Battery-36-Pack-E91SBP36H/100232088. [Accessed: 23- Feb- 2017].

[4] “Battery Myths vs Battery Facts”, 2017. [Online]. Available: http://www.greenbatteries.com/battery-myths-vs-battery-facts-1/ [Accessed: 20-Feb-2017].

[5] “Arduino Low Power - How to Run a ATmega328p For a Year On Coin Cell Battery”, 2017.[Online].Available:http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/. [Accessed: 21-Feb- 2017].

[6] “Pololu 3.3V Step-Up Voltage Regulator U1V10F3”, 2017. [Online]. Available: https://www.pololu.com/product/2563. [Accessed: 24-Feb-2017].

[7] “Debounce”, 2015. [Online]. Available: https://www.arduino.cc/en/tutorial/debounce [Accessed: 12-Mar-2017].

[8] “Omni-Directional Foil Electret Condenser Microphone”, 2010. [Online]. Available:

http://cdn.sparkfun.com/datasheets/Sensors/Sound/CEM-C9745JAD462P2.54R.pdf. [Accessed: 24- Feb- 2017].

[9] "Human Hearing and Voice", 2001. [Online]. Available: https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node271.html. [Accessed: 24- Feb- 2017].

[10] "Voice Level at Distance" [Online]. Available: http://www.engineeringtoolbox.com/voice-level-d_938.html [Accessed: 14-Mar-2017].

[11] Bowon Lee, Mark Hasegawa-Johnson, Camille Goudeseune, Suketu Kamdar, Sarah Borys, Ming Liu, and Thomas Huang, AVICAR: Audio-Visual Speech Corpus in a Car Environment. Interspeech, October 2004, pp. 380-383.

[12] Géron, Aurélien (2016). Chapter 4. Training Models - Gradient Descent. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. Retrieved from https://www.safaribooksonline.com/library/view/hands-on-machine-learning/9781491962282/ch04.html.

[13] Cortes, C., & Vapnik, V. (2014). Support-Vector Networks. Retrieved from http://homepages.rpi.edu/ bennek/class/ mmld/papers/svn.pdf

[14] "Low-Current, I²C, Serial Real-Time Clock". [Online]. Available: https://www.maximintegrated.com/en/products/digital/real-time-clocks/DS1339A.html/tb_tab0. [Accessed: 23- Feb- 2017].

[15] "Electret Mic Breakout Board Hookup Guide" [Online]. Available: https://learn.sparkfun.com/tutorials/electret-mic-breakout-board-hookup-guide

?_ga=1.8866480.46403813.1487651164#resources-and-going-further-
[Accessed: 14-Mar-2017].

[16] “Low Power, Single-Supply, Rail-To-Rail Operational MicroAmplifier™ Series (Rev. A)” [Online]. Available: http://www.ti.com/lit/ds/symlink/opa344.pdf [Accessed: 14-Mar-2017].

[17] Ieee.org, "IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Feb- 2017].

[18] acm.org, "ACM Code of Ethics and Professional Conduct", 1992. [Online]. Available:https://www.acm.org/about-acm/acm-code-of-ethics-and-professional -conduct. [Accessed: 08- Feb- 2017].