

Parents of the Future

By: Damian Komosa, Sasan Erfan and Yarshun Jayakumar

Final Report for ECE 445, Senior Design, Spring 2019

TA: David Null

01 May 2019

Project No. 32

Abstract

Chores are an area of society that lags behind the rest of the world when it comes to technological innovation and our invention aims to solve this problem. Parents of the Future is an invention that allows parents to have access to a web interface that tracks chore completion through the use of a hardware module that resides on a laundry basket or garbage can. The design features a modular design that includes a Sensor, Control, Power and WiFi subsystem, which together create a functioning hardware module responsible for tracking the status of a Laundry Basket or Garbage Can. Alongside the hardware module is a Web- hosted interface that allows the parents easy accessibility to months of relevant chore data and near- instant updates on chore completion responsibilities. The system is fully wireless through the use of a WiFi chip and features cost- efficient hardware that aim to innovate the Chore field.

Table of Contents

1. Introduction	1
2 Design	2
2.1 Control Module	2
2.1.1 Microcontroller	2
2.1.2 Status LEDs	3
2.2 Sensor Module	4
2.2.1 Ultrasonic Sensor	4
2.2.2 Force Sensitive Sensor	5
2.3 Power Module	6
2.3.1 Eight 1.5V AA Alkaline Batteries	6
2.3.2 5V Voltage Regulator	6
2.3.3 3.3V Voltage Regulator	7
2.3.4 Current Draw and System Battery Life	8
2.4 Wifi Module	9
2.4.1 Connecting the ESP8266-01 to WiFi	9
2.4.2 HTTP Post request and the REST API	10
2.5 Front- End Module	10
2.5.1 ThingSpeak	11
2.5.2 Web Server	11
2.5.3 Python script and Website	11
3 Design Verification	12
3.1 Control Module	12
3.1.1 Microcontroller	12
3.1.2 Status LEDs	12
3.2 Sensor Module	12
3.2.1 Ultrasonic Sensor	12
3.2.2 Force Sensitive Sensor	12
3.3 Power Module	13
3.3.1 Eight AA Alkaline Batteries	13
3.3.2 5V Voltage Regulator	13

3.3.3 3.3V Voltage Regulator	13
3.4 WiFi Module	13
3.4.1 WiFi chip to Database	13
3.4.2 WiFi chip to LAN	13
3.5 Front- End Module	13
3.5.1 Data Tables	13
3.5.2 Real Time Graph Updates	14
3.5.3 Web Page Components	14
3.5.4 Chore Notifications	14
4. Costs	15
4.1 Parts	15
4.2 Labor	15
5 Conclusion	16
5.1 Accomplishments	16
5.2 Uncertainties	16
5.3 Ethical considerations	16
5.4 Future work	17
References	18
Appendix A Requirement and Verification Table	19

1 Introduction

One of the biggest hurdles associated with chores is the fact that parents find it hard to keep track of its completion. Parents often find themselves going out of their way to check whether or not a certain chore has been taken care of. The act of physically checking the status of a chore can be tiring and is not something a person would like to go through after a long day at work.

The proposed solution aims to gather chore data placing it at the fingertips of the parents allowing ease of access. This will not only eliminate the need to physically move to check chore status but will also dismiss chore completion doubts that are bound to creep into the minds of parents. In order to accomplish this goal, we have chosen two critical sensors, the ultrasonic sensor and the pressure sensor, which will be indicative to determining a full or empty container. Our physical module will include an ATmega328P microcontroller to manage the sensor data and send it wirelessly using the ESP8266 WiFi chip to the online interface to be accessed by the parents. The whole system is powered by 8 AA Alkaline batteries which is then regulated to 5V and 3.3V depending on the requirement of each component. A database would be implemented to keep track of this sensor data. The website will not only display relevant sensor data and plots but will also notify the user every minute whether a specific chore needs to be completed. For the trash can to be classified as 'full', the thresholds set for each sensor has to be met. The ultrasonic sensor has to sense an obstacle within 12cm of the sensor and the pressure sensor has to sense a weight more than 450 units to be classified as 'full'. Upon completion of the technical parts of the project, the pressure sensor was attached to bottom of the trash can (to keep track of weight) while the ultrasonic sensor was attached to the top (to keep track of height). The project worked as expected and accurate sensor data was managed to be sent to the webpage, notifying the user each minute whether the chore needed to be completed or not.

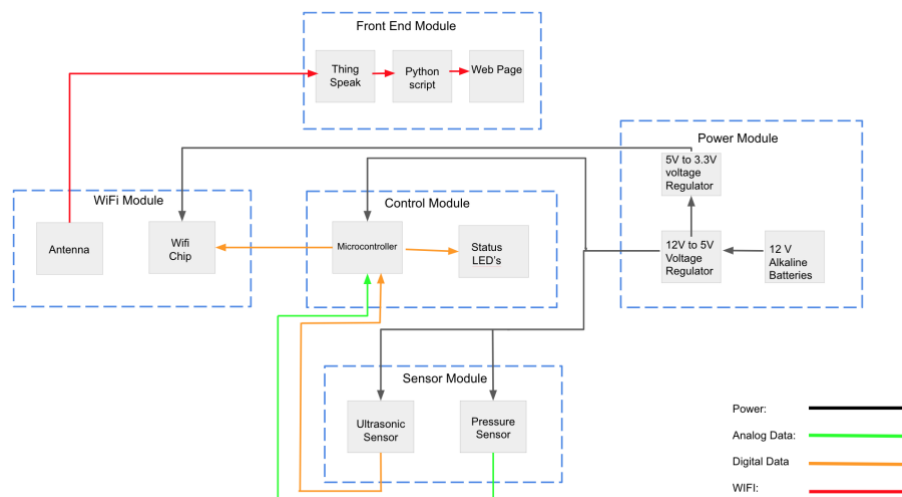


Figure 1. Block Diagram

2 Design

2.1 Control Module

The Control Module is responsible for the acquisition of data from the Sensor module and transmission of filtered data to the WiFi Module. Through the use of an ATmega328P microcontroller chip and Arduino Software, the analog and digital data from the sensors is properly formatted and manipulated in order to correctly represent the project environment. Additionally, the Status LEDs are used to indicate a properly functioning circuit and confirm when sensor thresholds are met, as indicated by our design decisions.

2.1.1 Microcontroller

The ATmega328P is the microcontroller chip that offered our project the most advantages because it offered the ability to use analog and digital data, allowed for the use of lower- power modes and the ability to use Arduino IDE. Additionally, the three to five-dollar price point made it a very affordable option when it comes to full- project marketability. The ATmega328P was one of the few options that we looked at which include: PIC32, STM32 and NodeMCU. The ATmega328P was chosen because it allowed for the use of a 5V input voltage, was well documented and provided the option of using low power modes. The ATmega328 has the ability to operate at clock speeds dependent on the crystal oscillator that's chosen and, in our case, we went with 16 MHz. The estimated Active Supply Current on the ATmega328P at 16 MHz with 5V is approximately 10 mA, which was confirmed during current testing.

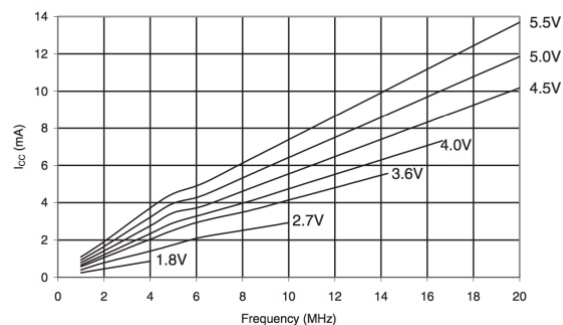


Figure 2. ATmega328P Active Supply Current vs. Frequency [2]

The software associated with the is largely associated with the transformation of digital and analog data to readable distance in centimeters and force in Newtons. Additionally, the specific connections for the Ultrasonic Sensor, Force Sensitive Sensor, ESP8266 WiFi Chip, Status LEDs and power inputs can be seen in the Eagle Circuit Schematic below. The software associated with the ATmega328P is also responsible for formatting sensors data into a specific string for the ESP8266 to send to the front- end module.

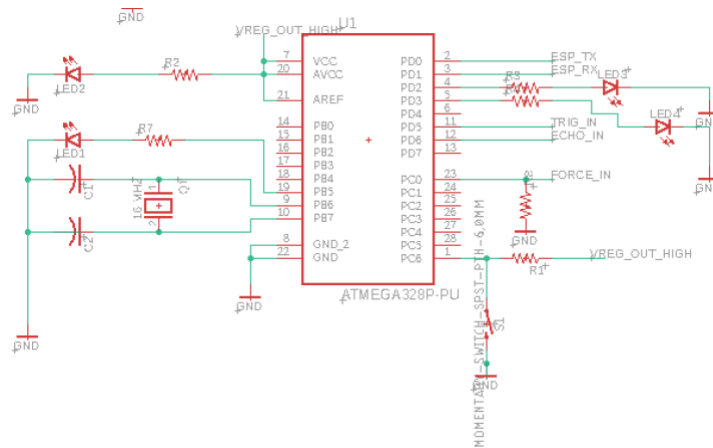


Figure 3. Control Module Eagle Schematic

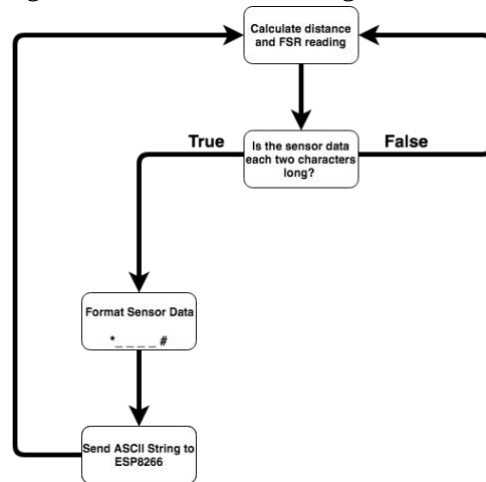


Figure 4. Control Module Software Flowchart

2.1.2 Status LEDs

The Status LEDs are an integral part of our design because they are responsible for assuring the customer that the device is receiving the correct amount of power from the power module. The LEDs we decided to use were standard 3mm, which were picked over the larger 5mm option in order to reduce current draw. The current testing of the LEDs revealed that they drew between 15 mA and 19 mA depending on the color of the LED.

Green LED - Indicates whether circuit is receiving 5V.

Yellow LED - Indicated when the ultrasonic sensor detects something within 12 cm.

Red LED - Indicated when the Force Sensitive Sensor detects weight.

Blue LED - Indicator LED for when both yellow and red LED conditions are met.

The Yellow, Red and Blue LEDs are controlled using digitalWrite command through digital pins on the ATmega328P. They reside on the PCB and are meant to represent the conditions set forth to determine a full- container command vs empty/ partially full.

2.2 Sensor Module

The sensor module is responsible for generating data that will be used to make chore completion decisions. The two sensors that were used in the design is the HC- SR04 Ultrasonic sensor and the Force Sensitive Sensor, which both check a different condition to account for any inaccuracies that could be caused by only having one sensor.

2.2.1 Ultrasonic Sensor

The HC- SR04 ultrasonic sensor will be used in our design for the sake of determining distance and aiding the pressure sensor in the conclusion of whether a container is full. The ultrasonic sensor works by emitting a 40 kHz ultrasound wave and because of the fact that sound bounces off objects, it will be able to record the time it takes for the wave to echo back. The ultrasound is generated when Trig is on the high state for 10 microseconds, which is followed by the sensor sending 8 cycles of sonic burst that bounce off the nearest object. The ultrasound travels at the speed of sound and after bouncing returns back to the Echo pin, which will record the time in microseconds that the wave traveled. This sensor is necessary in our design because if it were to detect an item within a specified distance of 7 in. for a series of 10 echo cycles then it will send a “Full” command to the ATmega328P microcontroller. The sensor will be receiving power from the voltage regulator and the TRIG and ECHO pins on the sensor will be connected to the PWM (Pulse Width Modulation) pins on the ATmega328P, which is digital input pin on the ATmega.

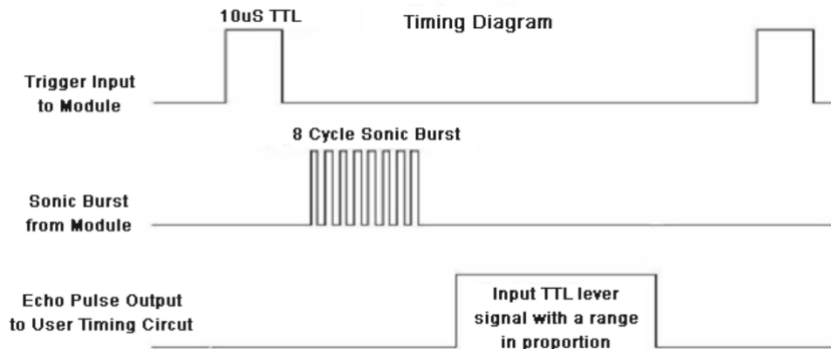


Figure 5. Ultrasonic Sensor Timing Diagram [1]

In order to use the ultrasonic sensor to measure the accurate distance there needs to be specific calculation implementation. Since the ultrasonic sensor will track the amount of time necessary for the pulse to travel from the sensor to the object and back, the equation must account for this part.

$$Distance = \left(\frac{Time}{2} \right) * Speed\ of\ Sound\ at\ Sea\ Level \quad (1)$$

The speed of sound that we are using will be approximately 340 m/s or 34000 cm/s since we wish to work with smaller magnitude measurements.

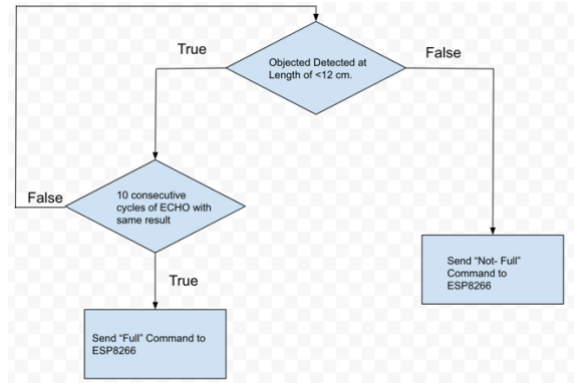


Figure 6. Ultrasonic Sensor Software Flowchart

2.2.2 Force Sensitive Sensor

The Force Sensitive Resistor, otherwise referred to as the pressure sensor will be used to determine whether there is weight detected in the respective container. This sensor can differentiate between objects weighing 100g all the way up to 10 kg through. Depending on the pressure applied to the sensor the resistance will vary from a magnitude of 0.5kΩ up to 50 kΩ and thus indicating the force applied on the sensor. The respective log- log graph for Force vs. Resistance is pictured below. This data will be useful in conjunction with the ultrasonic sensor to determine whether the container is full by determining a threshold of 4kg. One important detail for understanding the logic used to pick the weight categories associated with this sensor is the measuring area. Since the pressure sensor is only 2 x 2 inches in area, it won't be used to measure the full weight of the garbage, but instead a smaller portion of it. According to our research, an average garbage can contents weigh around 10 kg, so we have decided that detecting 2 kgs on our sensor would substantiate a 'full' command being sent. The sensor will be attached to 5V of power, which will be coming from the voltage regulator and the data will be sent to the Analog Input pin on the ATmega328P Microcontroller chip.

Table 1. Force vs. FSR Resistance [4]

Force (lb)	Force (N)	FSR Resistance	(FSR + R) ohm	Current thru FSR+R	Voltage across R
None	None	Infinite	Infinite!	0 mA	0V
0.04 lb	0.2 N	30 Kohm	40 Kohm	0.13 mA	1.3 V
0.22 lb	1 N	6 Kohm	16 Kohm	0.31 mA	3.1 V
2.2 lb	10 N	1 Kohm	11 Kohm	0.45 mA	4.5 V
22 lb	100 N	250 ohm	10.25 Kohm	0.49 mA	4.9 V

$$V_0 = V_{cc} \left[\frac{R}{R+FSR} \right] \quad (2)$$

The force calculation was estimated using equation two, which used a 10 kΩ pull- down resistor that was include for ease of calculation and the establishment of clean readings. Throughout the design, the threshold of 20 Newtons was used in order to establish the unit step response that is seen in the design of our project.

2.3 Power Module

The power module will contain the necessary components that will provide power to the ATmega328P, ultrasonic sensor, pressure sensor and the ESP8266. We decided on using eight AA alkaline batteries that are 1.5V each, providing 12V in total. This voltage value is suitable to be inputted to the LM7805 voltage regulator that requires at least 7V to receive a clean 5V output. We decided on eight AA batteries mainly due to simplicity and the fact that space is not an issue in our physical module.

2.3.1 Eight 1.5V AA Alkaline Batteries

The eight AA batteries would act as the input voltage needed to power sensors, the microcontroller and the WiFi chip. The type of battery that would be used is the Energizer Max AA Alkaline battery. The batteries connected in series would go through a 5V voltage regulator to convert the 12V input to a 5V output to directly power the sensors and the microcontroller. This 5V output would also go through a 3.3V voltage regulator which would go on to power the ESP8266 WiFi chip.

2.3.2 5V Voltage Regulator

The 12V input voltage from the eight AA alkaline batteries is not suitable to be fed directly to the microcontroller, sensors and the WiFi chip. The type of regulator used is the UA7805CKCT voltage regulator that needs a minimum input voltage of 7V to ensure a clean 5V output. This regulator would be used to make sure 5V is inputted to the sensors and microcontroller ensuring smooth operation within limits listed in the datasheets. The voltage regulator circuit would look as follows with the left nodes connected to the supply voltage of 12V:

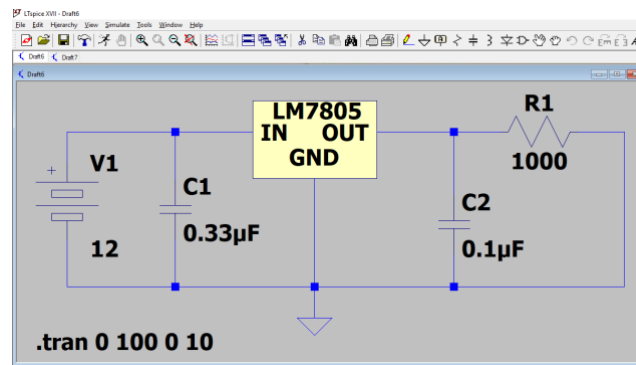


Figure 7. LTSpice circuit schematic of 5V voltage regulator

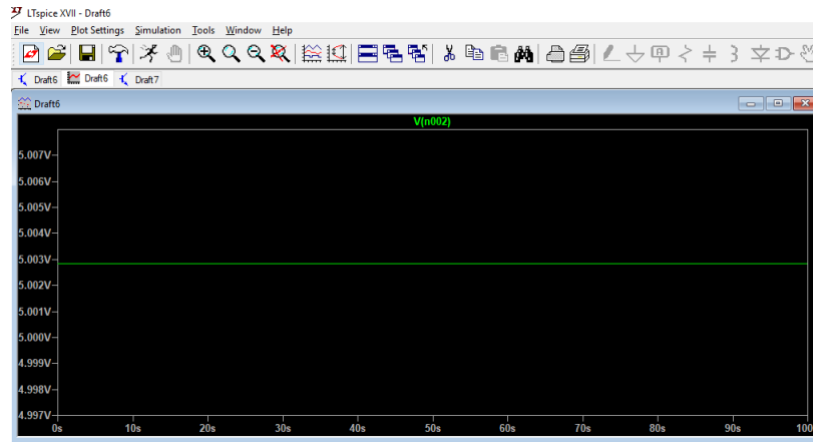


Figure 8. Output voltage obtained through LTSpice simulations

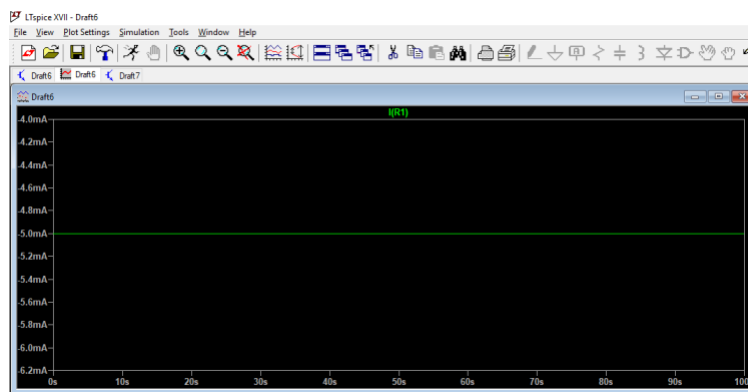


Figure 9. Current draw seen through LTSpice simulations

2.3.3 3.3V Voltage Regulator

The 12V output from the eight AA alkaline batteries in series would be too high to power the ESP8266 WiFi chip. Thus, another voltage regulator is used to step down the 5V from the 5V voltage regulator to 3.3V. The type of voltage regulator that will be used is the LD1117V33 voltage regulator. This decrease in voltage would ensure the voltage inputted to the ESP8266 WiFi chip is kept within suitable limits as listed in the datasheet. The 3.3V voltage regulator circuit would look as follows with the left nodes connected to a supply voltage of 5V from the 5V voltage regulator:

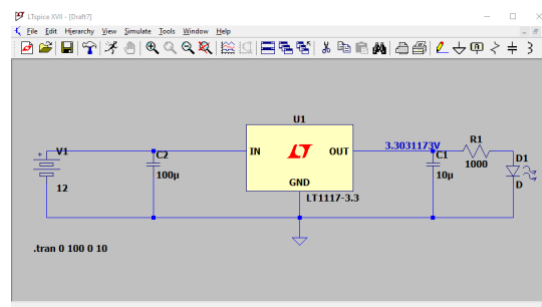


Figure 10. LTSpice circuit schematic of 3.3V voltage regulator

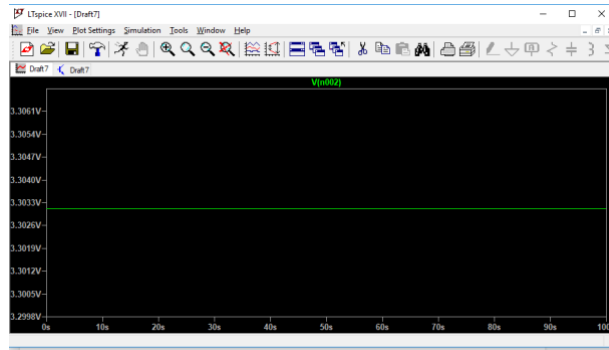


Figure 11. Output voltage obtained through LTSpice simulations

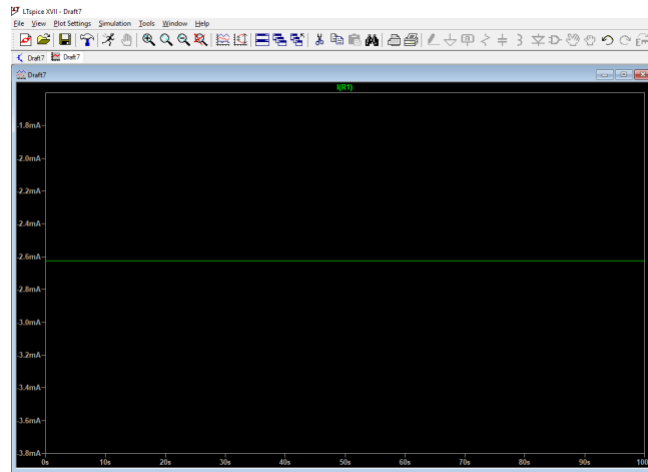


Figure 12. Current draw seen through LTSpice simulations

2.3.4 Current Draw and System Battery Life

Case 1: Current drawn by the ATmega 328P

Case 1a: When 5V is supplied to the ATmega 328P

Case 1b: When only the ultrasonic sensor threshold is met

Case 1c: When only the Force Sensitive Resistor (FSR) threshold is met

Case 1d: When both thresholds are met

Case 2: Current drawn by the Force Sensitive Resistor (FSR)

Case 3: Current drawn by the Ultrasonic Sensor

Case 4: Current drawn by the ESP8266

Table 2. Current Analysis

Type of Case	Current Drawn (mA)
1a	17.06
1b	28.20
1c	29.30
1d	48.20
2	0.505
3	3.070
4	70.00

About 60mA is drawn before the ESP8266 is connected and this amount rises to 130mA after it is connected. By taking into account the quiescent current draw on each voltage regulator, an additional 5mA is drawn from each regulator and this makes the total current drawn by the whole system about 140mA. With this value and the value of battery capacity, the battery life of the system can be computed as shown below:

$$\text{Battery Capacity} = (\text{Current drawn}) * (\text{Battery life}) \quad (3)$$

$$\text{Battery Life} = (2.5\text{Ah}) / 0.14\text{A} = 17.86 \text{ hours} \quad (4)$$

2.4 WiFi Module

The WiFi module is responsible for streaming sensor data to the home computer over a WiFi connection. The chip used to accomplish this is the ESP8266-01, which is unique in the WiFi chip space for being relatively well documented compared to alternative WiFi chips.

2.4.1 Connecting the ESP8266-01 to WiFi

Before we can connect the ESP8266 to WiFi, there are a few setup steps that need to be accomplished. The WiFi manager library, alongside the ESP8266 library need to be downloaded onto the Arduino IDE, and a functioning software serial connection with the AT Mega must be established. Without a functioning software serial connection, no data will be received by the WiFi chip and there will be no data sent to the computer.

The computer hosting the web server and the WiFi chip must be connected to the same network, and it is optimal to avoid the use of public networks for the sake of preventing interference. To have a network we can control, we set up a Local Area Network.

2.4.2 HTTP Post request and the REST API

As we will outline the section below, the location for the data sent by the WiFi chip is a database, and for our particular needs we use a technology called “ThingSpeak”. In order to transmit data across WiFi, a specific protocol needs to be adhered to, and in our case, we follow the HTTP protocol. HTTP protocol stands for “Hypertext Transfer Protocol” and is used in linking hypertext documents across the World Wide Web. HTTP supports two methods of data transfer with a text document on the internet: GET and POST. HTTP GET allows the user to ping a web page and retrieve data from it, and HTTP POST allows the user to ping a web page and send data to it.

In the context of our project, we want to send data to a cloud database hosted on the internet, so it is logical to use the HTTP POST protocol to send our data. There is another layer of complexity to this process, and that is introduced by REST API. REST stands for Representational State Transfer, and the methodology essentially expands upon HTTP’s base methods. For the cloud database we chose to use, a REST API was required to implement advanced data manipulation, such as real time graphs and Fourier Transform capabilities.

2.5 Front- End Module

The Front-End module is responsible for displaying all relevant data and notifying the user has a chore to complete. To accomplish this, the hierarchy can be seen in the following diagram:

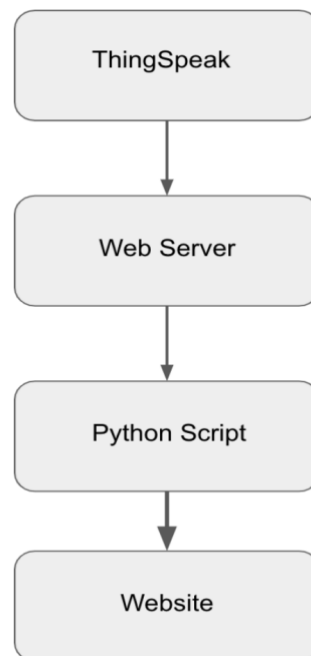


Figure 13. Front- End Module Software Flowchart

2.5.1 ThingSpeak

Our data must have an endpoint from the WiFi chip. That endpoint is the database, a server which can store tables of data with rows and columns. We chose ThingSpeak as our database because of its cloud storage capabilities and MATLAB functionality. Having a cloud hosted database means that the users don't have to worry about the application taking a large amount of storage on their computer, and helps data analytics become more streamlined since the graphing capabilities won't be relying on the user's computer.

2.5.2 Web Server

Now that we have a database containing our data, we need to begin hosting a website. In order to do this, we need to have a local web server running that is capable of accessing the local area network and supporting PHP functionality. For our web server, we used XAMPP, although there are a plethora of options that one could use. Alongside the web server's PHP capabilities, we also need to make sure the web server supports MySQL, since without a locally hosted database there is no way to keep track of who completed the chores or how many chores have been completed.

2.5.3 Python script and Website

With a database and a web server, the last remaining piece of the front-end is the website. The website is series of PHP linked documents, each containing access to a different part of the application. The Python script handles notifying the user about the status of whether there is a chore to complete or not. This is done via Python's BeautifulSoup module, which allows us to parse through the HTML tags of the ThingSpeak data to see what the value of each sensor is. We can then calculate whether there is a chore or not, and create a pop up notification to notify the user every thirty second.

As mentioned earlier, the website uses PHP to connect to the ThingSpeak database and graph the sensor data. This updates every minute, and you can think of each graph as an HTML document that is being printed out by the PHP script. PHP then makes a new graph after a new sensor data point is inputted, and then generates the HTML document that represents the new graph.

3 Design Verification

3.1 Control Module

The Control Module is required to act as a hub for data acquisition directly from the sensors and through programming must correctly filter data to be sent to the ESP8266. Additionally, the effective use of LEDs will allow the user to obtain real- time information on the status and proper- operation of the sensors. For details on the specific requirements and testing procedures, please see Appendix A.

3.1.1 Microcontroller

The Microcontroller requires that a voltage of 5V is provided in order to allow correct operation at 16 MHz. The Analog and Digital data obtained from the sensor module must be correctly translate to centimeter values for the ultrasonic sensor and newton values for the force sensitive sensor. Additionally, the sensor data must be correctly filtered and formatted before being sent to the ESP8266 WiFi chip over serial communication pins. For details on the specific requirements and testing procedures, please see Appendix A.

3.1.2 Status LEDs

Through the use of four LEDs, the customer will be able to obtain crucial information regarding proper voltage input and correct sensor operation. The four LEDs must correctly display: whether the module is receiving correct input voltage of 12 V and whether certain thresholds for the ultrasonic and force sensitive sensors are being interpreted correctly. For details on the specific requirements and testing procedures, please see Appendix A.

3.2 Sensor Module

3.2.1 Ultrasonic Sensor

The HC- SR04 Ultrasonic sensor is responsible for detecting clothes or garbage at the top brim of a laundry basket or garbage can through the use of distance- finding capabilities. This means that it must accurately detect distance within a range of 50 cm., which was done by creating an algorithm for this calculation. For details on the specific requirements and testing procedures, please see Appendix A.

3.2.2 Force Sensitive Sensor

The Force Sensitive Sensor is responsible for updating the Control Module on whether a weight threshold was surpassed. The calculation of force in Newtons was required for this weight threshold and this was done through programming voltage and resistance fluctuations when force was applied. For details on the specific requirements and testing procedures, please see Appendix A.

3.3 Power Module

3.3.1 Eight AA Alkaline Batteries

The batteries are responsible for making sure 12V is supplied to the 5V voltage regulator. A minimum of 7V is needed to make sure the regulator outputs a clean 5V. For details on the specific requirements and testing procedures, please see Appendix A.

3.3.2 5V Voltage Regulator

The 5V voltage regulator is responsible for making sure 12V is correctly regulated to 5V which will then power the ATmega328P, ultrasonic sensor and pressure sensor. For details on the specific requirements and testing procedures, please see Appendix A.

3.3.3 3.3V Voltage Regulator

The 3.3V voltage regulator is responsible for making sure 5V is correctly regulated to 3.3V which will then power the ESP8266 WiFi chip. For details on the specific requirements and testing procedures, please see Appendix A.

3.4 WiFi Module

3.4.1 WiFi chip to Database

The WiFi chip is responsible for correctly streaming data to the database, and without this critical component the application will fall apart. For details on the specific requirements and testing procedures, please see Appendix A.

3.4.2 WiFi chip to LAN

The WiFi chip must be able to connect to a local area network, and on a general note must be able to properly connect to any WiFi network of the user's choice. For details on the specific requirements and testing procedures, please see Appendix A.

3.5 Front- End Module

3.5.1 Data Tables

The website must be able to show integer values of what the sensors are currently detecting via parsing the ThingSpeak database. For details on the specific requirements and testing procedures, please see Appendix A.

3.5.2 Real Time Graph Updates

The graphs of sensor data on the web page must update every 5 minutes, and must update with the correct time stamped value of the sensor data point. For details on the specific requirements and testing procedures, please see Appendix A.

3.5.3 Web Page Components

The web page must display two graphs and two tables for each of the respective sensors, and the web page run without errors on the local web server. For details on the specific requirements and testing procedures, please see Appendix A.

3.5.4 Chore Notifications

A pop-up notification must occur every thirty seconds, notifying the user whether or not there is a chore to be completed. This functionality is implemented via a Python script performing web scraping on our own web site, and must have a functioning GUI. For details on the specific requirements and testing procedures, please see Appendix A.

4 Costs

4.1 Parts

Table 3. Parts Costs

Part	Manufacturer	Retail Cost (\$)	Bulk Cost (\$)	Actual Cost (\$)
WiFi Module – ESP8266	Sparkfun	6.95	6.26	6.95
ATMEGA328P	Microchip Technology	2.14	1.78	2.14
Force Sensitive Resistor	Sparkfun	11.25	10.13	11.25
Ultrasonic Ranging Module	Sparkfun	3.95	3.56	3.95
UA7805CKCT Voltage Regulator	Texas Instruments	0.58	0.21	0.58
AA Alkaline Batteries (8x)	Energizer	4.69	3.42	4.69
2 10uF capacitors, 2 1uF caps, 3 10kOhm resistors, 1 220Ohm resistor	Mouser	6.00 (Estimated)	3.00 (Estimated)	6.00 (Estimated)
16MHz AT49S 20PF DIP Quartz Crystal Oscillator	uxcell	0.098	0.098	0.098
LD1117V33-DG	Mouser	0.91	0.35	0.91
LEDs (4x)	Plusivo	0.12	0.12	0.12
Total		36.69	28.93	36.69

4.2 Labor

Table 4. Labor Cost Analysis

Engineer	Rate ($\frac{\$}{hr.}$)	Hours	Total (\$)
Damian Komosa	35	250	21, 875
Sasan Erfan	35	250	21, 875
Yarshun Jayakumar	35	250	21, 875
Total		750	65, 625

5 Conclusion

5.1 Accomplishments

The project was overall a massive success with the modified high-level requirements being met during the live demonstration. The project was successfully able to collect and filter data, while providing full wireless connectivity with the HTML website on the front-end module. The HC-SR04 ultrasonic sensor provided accurate and timely distance information, which provided data partially responsible for determining a 'FULL' garbage can/ laundry basket vs. an 'EMPTY' one. The force sensitive sensor worked in a binary way where there were only two possible outputs for the reading, but it helped determine and conclude whether a container was indeed 'FULL.' The ultrasonic sensor and the force sensitive resistor helped determine chore completion status and the webpage beautifully displayed this data using integrated MATLAB data visualization.

5.2 Uncertainties

Although the project accomplished the goals set out by the high-level requirements, there was a slight amount of uncertainty when it came to the results obtained from the force sensitive resistor. The force sensitive resistor operated on a log-log scale where the resistance varied logarithmically with the force, which was the value we were originally planning to work with. The percentage error between the actual force applied and force calculated using the Arduino software was significant enough to where it was decided that we will only observe the general range of values. At this point it was decided that using a unit step function in order to determine the threshold of the force sensor would be more accurate and provide more utility than the exact value of the force.

5.3 Ethical considerations

Any future work on this project will have two primary ethical concerns to consider. The first one concerns safety and being electrical engineers, we are expected to abide by the IEEE Code of Ethics and Code #1 states that we agree "to hold paramount the safety, health and welfare of the public" [3]. The current components are currently not housed in a physical module and this makes it exposed. The fact that alkaline batteries and wires are exposed makes it susceptible to water damage and interference from little kids. Both of these external factors pose a risk and by encasing the wiring and circuit board, this risk can be minimized. Since the design involves a laundry basket and trash can, special arrangements regarding water do not have to be made. However, it is important to be aware of such precautions in case wet hands or wet clothes were to make contact with the battery or physical circuitry in general.

The second ethical concern revolves around data acquisition. We are responsible for the data collected indicating whether the chore needs to be done and it has to be accurate to ensure the integrity of this project. This is in accordance to Code #3 in the IEEE Code of Ethics that states that we agree "to be honest and realistic in stating claims or estimates based on available data" [3]. Tests will be conducted to make sure sensors and microcontroller accurately collect and send data to the database for it to be viewed by the user.

5.4 Future work

With all being said and done, our project has a great foundation we can build upon towards a great future. Our next step would be to create a mobile app with push notifications, as it would be far more convenient for the users to see chore notifications on their phone rather than lugging around their computer everywhere. Following that priority, we would then move towards creating a profile system, so that each child in the household could have their own account and thus have their own chores. We would then have a decision tree of work we could complete in any order: a ranking system, a universal leaderboard for all who own the product and expanding to AWS cloud services for faster service. There are also design we could make, such as creating a chassis for the product or changing the style of the website. All in all, there are lots of advancements to be made on our exciting product.

References

- [1]. Cdn.sparkfun.com. (2019). [online] Available at: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> [Accessed 1 May 2019].
- [2]. Ww1.microchip.com. (2019). [online] Available at: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf> [Accessed 7 Mar. 2019].
- [3] Ieee.org. (2019). IEEE Code of Ethics. [online] Available at: <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed 08 Feb. 2019].
- [4]. Learn.adafruit.com. (2012). *Using an FSR | Force Sensitive Resistor (FSR) | Adafruit Learning System*. [online] Available at: <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr> [Accessed 1 May 2019].

Appendix A Requirement and Verification Table

Table 5. R& V Table

Requirement	Verification	Verification (Y or N)
ThingSpeak must be able to accept 4 bytes of data from WiFi chip.	ESP8266-01 should send a 4-byte value to database and the database should correctly update. This can be reflected in the ThingSpeak channel.	Y
ThingSpeak must be able to transmit 4 bytes of data to the web page.	ThingSpeak should be shown on the web page. This can be checked by viewing the web page on the web server.	Y
ThingSpeak must be able to display a sensor data vs time graph, updating every 5 minutes.	The graph must update every 5 minutes. This can be checked by viewing the ThingSpeak data channel. This is a minimum requirement.	Y
The web page must be able to display two graphs for each sensor, and the individual value of each sensor.	The web page can be viewed via the web server, and must be able to display all four items listed.	Y
If a chore can be completed, there must be a notification to the user that it is time to complete a chore. This must refresh every minute. The notification must also describe whether there are no chores to complete.	This can be tested via the Python module and the PCB transmitting data to the database. A pop up should appear for a chore that can be completed or cannot be completed every minute.	Y
The WiFi Chip must be able to connect to a specific WiFi Network, which can be a hotspot or router.	This will be verified on the iPhone, which will indicate the number of devices connected to the Person Hotspot. Without the laptop connected, the phone should indicate that 1 device is connected to the network.	Y
The 12V from the battery has to be converted to 5V when inputted to each sensor and the 3.3V regulator.	A voltmeter can be used to make sure the output pin (right pin) of the 5V regulator is at a value of $5V \pm 0.5V$.	Y
When the power module is connected to the PCB circuit, the green LED being powered on will indicate that the sensors and ATmega are receiving 5V.	A voltmeter can be used to make sure the output pin (right pin) of the 5V regulator is at a value of $5V \pm 0.5V$. The green LED will also turn on.	Y

The 5V from the 5V voltage regulator has to be converted to 3.3V when inputted to the WiFi chip.	A voltmeter can be used to make sure the output pin of the 3.3V regulator is at a value of $3.3V \pm 0.3V$.	Y
When 3 LEDs are on and the ATmega is powered on, that respective circuit should be drawing a maximum of $60mA \pm 20mA$.	The current is measured by placing an ammeter in series with the input of the ATmega to make sure the outputted current is within the proper region of $60mA \pm 20mA$.	Y
The ATmega328P must be receiving 5V from the Voltage Regulator on the Power Module. There will be an allowed error of $\mp 0.5 V$ in order to account for any voltage instability in the Voltage Regulator.	This will be verified using a standard voltmeter and these tests will be done on Pin 7 of the ATmega328P, which resides on the PCB.	Y
When the ATmega is provided an input voltage of 5 V and the frequency being 16MHz, the supply current should be approximately around 13- 25 mA with a percent error of 5%.	This will be measured using a digital ammeter on an external breadboard that has the copy of the PCB layout. The measurements will be done between the output of the 5V voltage regulator and power input of the ATmega, which is pin 7.	Y
The microcontroller will generally be used in a typical household environment so it must be able to perform data acquisition and remain fully functional in temperatures ranging from 50 degrees to 90 degrees Fahrenheit	This will be tested through the confirmation of full functionality throughout the application of external heat and cooling. The ATmega328P will be kept in their conditions for a period of 30 minutes and tested afterwards.	Y
When the ATmega is correctly powered with an input voltage of 5V and receiving correct data from the Ultrasonic/ Pressure sensor the respective LEDs should power on.	When the power module is connected to the PCB, the sensors are connected to ATmega, the LED should be held in the high position. This will be verified through the use of a multimeter	Y
The ultrasonic sensor must be able to measure distance using the implemented algorithm and sense when an object is detected within 12 centimeters.	This sensor will be tested by checking that there is a separate response when something is detected at a distance >12 centimeters versus <12 centimeters. This can be seen by observing the Yellow Status LED.	Y

The Ultrasonic Sensor should be receiving 5V from the Voltage Regulator in order to provide ample voltage for correct operation.	A voltmeter can be used to make sure a value of $5V \pm 0.5V$ is inputted to the Ultrasonic Sensor.	Y
The ultrasonic sensor should not be drawing in excess of 5mA of current.	The current is measured by using a digital ammeter in series with the input terminal of the Ultrasonic Sensor. A value of at least $3mA \pm 2mA$ should be recorded.	Y
The sensor must be programmed to respond differently to weight lower than 450 units vs. detected weights of greater than 450 units	This will be tested by applying various weights lower than 450 units and making sure that the sensor isn't triggering a full command. The same will be done to weight higher than 450 and determining that a full command is triggered. This can be verified using the Red Status LED.	Y
The Force Sensitive Resistor should be receiving 5V from the Voltage Regulator in order to provide ample voltage for correct operation.	A voltmeter can be used to make sure a value of $5V \pm 0.5V$ is inputted to the Force Resistive sensor.	Y
The Force Sensitive Resistor should not be drawing in excess of 1mA of current.	The current is measured by using a digital ammeter in series with the input terminal of the Force Resistive Sensor. A value of at least $0.5mA \pm 0.5mA$ should be recorded.	Y