# Solar Battery Powered Awareness of Closeness Kit - Solar BatPACK

Rafael Cejas, Marcelo Farias, Gabriela Gandorphi, Leticia Maia, Pedro Resende
IIT - Illinois Institute of Technology
Advisor: Professor Jeremy Hajek

## Abstract

This project was designed to help people with tunnel vision. These people don't have any peripheral vision and because of that they struggle with daily tasks. Our work was focused in improve a former system that addressed this issue. The system used ultrasonic sensors and vibration motors placed on a hat. Our team redesigned the prototype, changing the sensors and decided to use a backpack. The backpack was chosen because it would be more practical and would have a better performance than a hat. After changing the sensors we optimized the code to solve a couple of problems such as the serious response delay issue that the previous group had. An energy analysis was also made and a second solar panel was added to the new prototype. The electrical consumption was analyzed and optimized, allowing a full day of operation without the need of being connected to the grid. With all the optimizations and some minor design alterations, it is hoped that this project can evolve to an industrial level and, therefore, meet with the needs of any vision impaired person.

*Keywords:* Tunnel Vision, Solar Power, Arduino, Ultrasonic Sensors, Wearable.

Solar Battery Powered Awareness of Closeness Kit

## Introduction

Tunnel vision is an eye disease in which a person loses his or hers peripheral vision. Having tunnel vision is like looking through a binocular or telescope. Without their peripheral vision, people become more susceptible bumping into things or people on their side. That causes them to also be more susceptible to injuries from the collisions. The idea was having a device that warns a visually impaired person when there is something or someone nearby, and the general direction where they are.

## Previous Project

Our project is the continuation of a project started by another team in a previous semester. The previous team assembled a prototype in a hat. The hat had three sensors facing the sides and the front of the person wearing the hat, and three vibration motors in the same three sides. The motors vibrated when the corresponding sensors recognized an object up to 70 cm (2.2 feet) from the hat. Our job was to debug and optimize their work. The previous prototype was not left intact, so the first thing we had to do was rebuild it in a protoboard, and run the other group's code to see if it works. Afterworks, we started working on the old prototype: testing, debugging and improving.

The circuit and operation were relatively simple. Between the solar panel and the processor (Arduino Micro) there were three components: a solar lithium ion/polymer Charger; a 3.7V 1200 mAH lithium ion polymer battery and a PowerBoost 500 (5V USB Boost). The solar panel was connected with the Charger, which decreased the voltage to 3.7V (safe to the battery). Then, the PowerBoost increased the voltage to 5V (compatible with the Arduino Micro). The three ultrasonic sensors - PING))) model - had 3 connections each: 5V, ground and signal, and each vibration motor (AdaFruit Mini Motor Disc) had ground and signal connections. The signals from these devices were all connected to digital pins from the Arduino, and the 5V and ground connections were attached to the PowerBoost correspondent pins. To determine the program status (on or off) the system had a touchpad with two connections. One was attached in an analog pin from the Arduino (so it would be able to sense an intense or soft touch) and the other to ground.

The Arduino, with the code uploaded in it, coordinated the operation. It was sending the sensors' pulse, receiving it back and evaluating the elapsed time (thus, the distance). It also calculated a mean with the last 20 distance values (filtering) and verified if this mean was within the 70 cm range. Finally, it vibrated the motors accordingly with the averaged distance. Parallel to this, the code was continuously checking the touchpad status.

A diagram from the previous circuit can be seen in Figure 1.

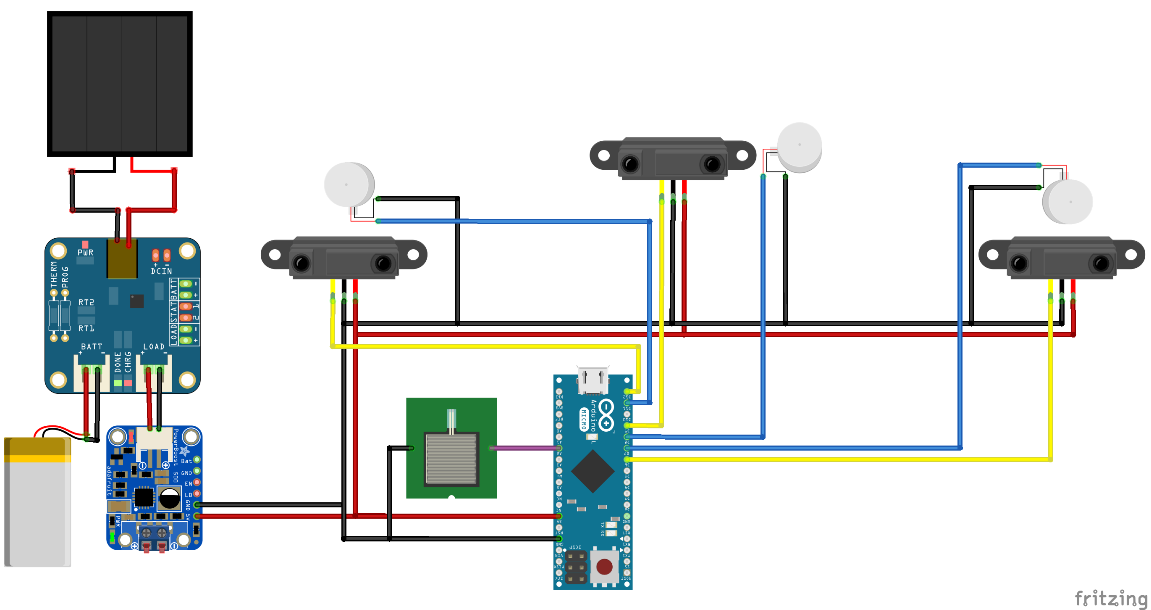

*Figure 1.* Previous Project Circuit.

## Proposed Changes and Optimizations

### New Range

An alteration considered was regarding the detectable range set by the code. The previous code had a maximum allowed distance of 70 cm, meaning that any obstacle located further than this value would be considered too far. After observing some everyday situations, we decided to increase the maximum allowed distance. We sensed that 70 cm wasn't enough to give the user time to feel the object's presence and deviate from it (considering an obstacle or person coming to the user's direction). The new maximum value set was 2 m.

### Change On/Off Mechanism

One of the first things that we thought about was the on/off mechanism. The previous group used a flexible touchpad, but they had some problems with it. They were not able to handle well long presses to the touchpad. If a person kept pressing the touchpad for a long time, the system would keep switching on and off. In addition, a person could easily press the touchpad by accident just by holding the hat's panel. That could cause confusion, because there is no way to know if the system is on or off without vibrating the motors. Therefore, one solution presented was to change the flexible touchpad for a simple button.

### Change of Ultrasonic Sensors

We also considered to replace the existing three PING))) ultrasonic distance sensors (Figure 2a) for LV-MaxSonar-EZ1 sonar sensors (Figure 2b) that were available in the laboratory.

There were two main reasons to consider this change. The first one was based on a primary reading of both data-sheets. It was stated that not only the LV-MaxSonar sensors were great for people and motion detection (it can detect people up to 2.4m distant), but also that the PING))) sensors couldn't detect soft surfaces accurately (such as clothes). The idea of the project wasn't restricted to people detection only, but since both sensors were expected to perform similarly well with solid objects, the people detection (and, hence, cloth detection) was a key parameter to consider.

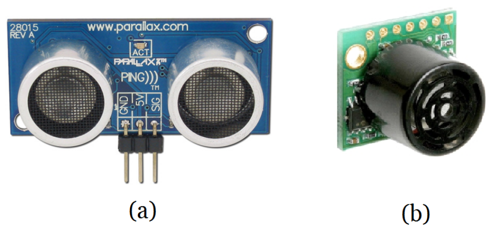


*Figure 2*. Ultrasonic sensors: a) PING))); b) LV-MaxSonar EZ1.

The second motivation for the change was mostly aesthetic. Considering that the system would have to be placed close to the user's body (and visible to others), the aspect and size of the sensors were important features. As can be seen in Figure 2, The LV-MaxSonar-EZ1 model is much smaller and discrete than the PING))) one. Therefore, along with the first reason mentioned, the LV-MaxSonar-EZ1 sensors were the preferred ones for the new project.

An interesting note worth mentioning is regarding the detectable distance values of the models. The PING))) sensors data-sheet presented a range between 2 cm and 300 cm, while the LV sensors specifications stated that they could produce distance measurements from 15.2 cm to 645 cm. Despite the relatively large difference between both models, the range wasn't a main factor for the decision. Since it was determined that a detection up to 2 m was satisfactory, both sensor models were adequate for the purpose.

Several tests were performed to confirm all those specifications, and the results are presented later in this report.

## Code Optimization

One of the problems that were presented to us at the beginning, was a delay in vibration, that could be related to a slow code. One thing we noticed that could be optimized was the loop to calculate the mean of the distance values every time a new value was added. The new mean could be simply calculated by subtracting a value and adding another.

Regarding the optimization of energy consumption, we thought of implementing a sleep function. We would put the processor to sleep while the system was off, reducing the number of used processor cycles, and, therefore, reducing the energy use.

## Change in Vibe Motors' Output Mode

The previous group had devised a way to relate the proximity of an object to a certain kind of vibration interval. As an object got closer, the interval between vibrations would decrease, and that way the person would have an idea of their distance to the object. We discovered that the sensors could be used with analog output (0-255 instead of HIGH/LOW) and thought that it might make the program simpler if we, instead of relating distance to pause between vibrations, related it to intensity of vibration. That way, as an object got closer, a person would feel a stronger vibration.

## Move System to a Backpack

The prototype in a hat made sense because it was easy to put on and remove, but we found some problems with it, such as:

- Carrying all the weight of the system on the head;
- Feeling the vibration on the head might be uncomfortable;

- The height that the sensors were placed made it harder to identify some obstacles.

At first, It was suggested that we rebuild the prototype in a shirt, because that would help with the issues listed above. However, other problems came up. The shirt would not be washable, and It would not be easily removable.

Improving the idea, we thought about a backpack, with a pocket to hold the system, the solar panel spread in the back, one sensor and vibration motor in each strap, and one sensor and vibration motor in the back. After we built a rough prototype, we thought It would be better to add a strap around the waist to keep the backpack in place. We also moved one vibration motor from the back to this strap, because It was easier to feel the vibration there.

### Add Second Solar Panel

Given the previous prototype, our main idea was to perform a complete energy analysis in order to fully understand the energy dynamics of the system. However, even before running any tests, we had some ideas that included the possibility of adding a second solar panel to the project. Since we decided to move the system from the cap to the backpack, the additional space would make it possible to add a new panel and therefore increase our power production.

## Preliminary Tests and Results

### Previous Prototype

After thinking and brainstorming new ideas and possible changes to the project, we started the actual tests in order to further analyze the energy consumption and production of the previous prototype. First, we decided to test all components separately, starting with a voltage test on the solar panel. The next step was to check the other components by creating an appropriated circuit for each one of them.

Once we knew that all the components were working properly, we moved on to the energy consumption analysis. The analysis consisted on using the multimeter to measure the current at different places (current leaving solar panel, current arriving/leaving battery, and current leaving PowerBoost/arriving system) and situations (sunny day and cloudy day). All the tests were made on IIT campus, during the summer, and from 10AM to 3PM, so we could have a good range of solar insolation. The values measured are located on Tables 1 and 2 and each situation is shown.

During our tests, we realized that the system would consume a significant amount of energy even when it was turned off. In the case of having the battery as the only source of energy, the current leaving the PowerBoost with the system turned off was about 55mA (Table 1).

Table 1

*Current measurements in different part of the system, for different situations*

| | Without Solar Panel | | | | | |
|---|---|---|---|---|---|---|
| | Current Leaving PowerBoost [mA] | | Current in each Ultrasonic Sensor [mA] | | Current in each Vibration Motor [mA] | |
| Situations | Only Battery | From Computer | Only Battery | From Computer | Only Battery | From Computer |
| Button Off | 55.0 | 42.0 | 6.2 | 5.9 | 0.00 | 0.00 |
| Button On, Without Obstacles | 69.0 | 57.0 | 12.0 | 12.0 | 0.02 | 0.05 |
| Button On, Closest Obstacles | 170.0 | 121.0 | 8.0 | 7.9 | 45.0 to 49.0 | 45.0 to 49.0 |

Besides that, we estimated the consumption of the battery, measuring the current leaving it in different scenarios, these different situations helped us to estimate the energy necessary for a normal day of use (Table 2).

After calculating the energy consumption and the usage of the battery, we decided to analyze how one solar panel would affect the situation. To evaluate that, we ran the battery tests again, first using a reliable source of electricity (a laptop via USB) and then using a solar panel during different insolations. As we expected, the laptop provided a constant current to the battery while generating energy for the system in different situations as it can be seen also in Table 2.

Table 2
*Current measurements into the battery, for different situations*

| | Current Into the Battery; System Attached | | | |
|---|---|---|---|---|
| | Without Solar Panel; Nor Computer [mA] | Without Solar Panel; With Computer [mA] | With One Solar Panel [mA] | |
| Situations | | | Sunny Day | Cloudy Day |
| Button Off | -90 | 43.9 | 55 | -72 to -85 |
| Button On, Without Obstacles | -92 | 44.1 | 32 | -119 |
| Button On, Closest Obstacles | -350 | 44.3 | -195 | -400 |

On the other hand, the solar panel proved to be much more inefficient. We already knew before starting that our flexible solar panel would generate less than 167 mA at its best, due to its specifications. During our experiments, we concluded that the solar panel would charge the battery only in two situations, both in a sunny day: when the system was off or when it was on but with no obstacles in its surroundings. In all other situations the solar panel wouldn't provide enough energy to the system, and consequently would only slow down the battery's discharge. The numbers were really relevant and we discovered that during a cloudy day, the system would drain the battery even if the button was off.

**Battery Tests**

After acquiring these data we decided to run a different test. We tried to confirm how long would take to fully discharge our batteries. First we used the battery specifications, the measured current and voltage to calculate how long it should take to drain the battery. Doing that, we estimated that it would take around 5 hours to discharge the battery while the system was detecting an obstacle at the closest range. We also calculated that It would take around 12 hours to discharge it while the system was on but not detecting anything, and around 13 hours with the button turned off.

To confirm our calculations we ran the following test. First, we fully charged our battery to be sure we had it maximum charge. Then, we connected it to the system in the three different situations described. Our empirical results were: 4.5 hours to drain the battery when an obstacle was placed right in front of the sensors, around 10 hours with the button on without obstacle and around the same for when the system was off.

**Second Solar Panel**

Following the decision of adding the second solar panel, we started running tests again, now with two solar panels attached to the system. This decision was made in order to increase the energy production, and therefore improving our battery life and the system's efficiency as a whole. Our tests consisted on measuring the current leaving the set of two solar panels and the current arriving/leaving the battery at the same conditions as the previous tests. As we can see in Table 3, the results were in general as good as expected and in some cases, even better than so. The only curious result was the current going to the battery on a sunny day with obstacle close to the sensors. With one panel the current was leaving the battery with 195 mA, but with 2 panels the battery was being charged with 65 mA. This represents an increase of 133%.

Table 3

*Current into the battery, with two flexible solar panels, for different situations*

| | Current Going to the Battery, With Two Flexible Solar Panels (System attached) [mA] | |
|---|---|---|
| Situations | Sunny Day | Cloudy Day |
| Button Off | 155 | -57 |
| Button On, Without Obstacles | 153 | -70 to -76 |
| Button On, Closest Obstacles | 65 | -350 to -370 |

**Hard Solar Panel**

Another solution for this problem was to change the solar panel, since there are several types of solar panels and technologies that can provide enough energy for our system. One example was the Medium 6V 2W solar panel from Adafruit, which was used in one of our prototypes. This panel had a higher output than the two flexible ones combined. However, after some tests, we decided to stick with the two flexible panels because of the effect of shading in both solutions. If a shadow was placed on only 1/6 of the new solar panel, its output current would drop from 150 mA to 20 mA. On the other hand, to make the flexible solar panels drop to 20 mA it was necessary to place a shadow upon 1/2 of their area. Since we expect that the users will face several shadows upon the solar panels, we decided to use the solar panels that had a better response to shading, the two flexible ones.

### Sleep Function and Optimization in Energy Consumption

The sleep function is triggered when the Arduino starts or when the button is pressed. This function allows the Aduino to stay on without running the code, at a low energy cost. When the button is pressed, the vibration motors and the sensors are turned off and the Arduino sleep function is executed. We set the button pin as an interrupt pin that wakes up the Arduino when the button is pressed. After waking up, the Arduino runs the end of the sleep function, which will enable the vibration motors and sensors again. This optimization resulted in a big improvement in terms of energy use. The current from the PowerBoost to the system before the addition of the sleep function, when the button was off, was 55 mA. After we added the code to put the system to sleep, it became 26 mA. This is about a 45% improve in energy use.

In the search for a greater reduction, we used a transistor to block the current to the sensors while the Arduino slept. This helped saving even more energy, because only the Arduino remained on while the system was off, using only 7 mA.

Despite these great results, by replacing the button for a switch between the Charger and the PowerBoost, we reduced the amount of energy required by the system to negligible values. Measuring the current leaving the battery we found 28.3 $\mu$A, probably due to the Charger circuit. With this value, besides the amount of energy saved, the battery can even be charged with low insolation.

### Ultrasonic Sensors

Before implementing the LV-MaxSonar-EZ1 sensors into the system, some tests were performed to support the decision, and also to validate the information given in both data-sheets.

**Single Sensor Tests.** For the first tests, we used a simple code that was just sending the pulse and evaluating the distance. With this code and only one sensor of each model, some interesting results were noticed.

For both models, the sensors detectable distance range showed to be accordingly with the specifications stated in the data-sheets, as presented in Table 4.

Table 4
*Detectable Distance Range for both Models*

| | Sensor's Model | | | |
|---|---|---|---|---|
| | PING))) | | LV-MaxSonar EZ1 | |
| | Laboratory Test | Data-Sheet | Laboratory Test | Data-Sheet |
| Minimum Distance | 3 cm | 2 cm | 20 cm | 15.2 cm |
| Maximum Distance | 311 cm | 300 cm | 635 cm | 645 cm |

The PING))) sensor presented inaccurate measures when trying to detect clothes. For short periods of time, with a person standing just in front of the sensor and without moving, the sensor outputted its maximum detectable distance value (311 cm). A probable explanation is that the emitted pulse was being absorbed by the clothes and thus, since it wasn't returning or taking much longer to do it, the program was interpreting that the object was at the furthest location possible. The LV-MaxSonar presented excellent results with clothes, with just rare cases of imprecise values.

We also observed that the LV-MaxSonar-EZ1 model was very sensible to its connections. An intense handling of the cables caused the distance measured values to sometimes reach zero. Furthermore, when this situation occurred, the program's loop time increased significantly.

Both sensors showed extremely fast responses in this simple situation, without significant delay of any kind.

**Complete System Tests.** After these observations, for each model, the system was assembled with the three sensors needed, and the previous project's working code was uploaded for more tests.

Regarding the detection of soft objects, such as clothes, the system with the LV sensors provided accurate distance values. On the other hand, when PING))) sensors were used, the measurements were, as expected, very imprecise, causing the vibe motors to wrongly vibrate or, sometimes, not vibrate at all.

With the three LV sensors, the same sensibility problem regarding connections was identified. At first, it was confused with an interference issue, since three sensors were in use, and the zero distance values were happening much more often. However, after several trials and tests the cables influence became clear when by only handling the sensors' connections, the accurate values reappeared.

With the three PING))) sensors, the system didn't present any significant response delay. On the other hand a relatively long response delay was noticed with the LV-MaxSonar-EZ1 model. The time taken for a vibe motor to vibrate after placing an object in front of the correspondent sensor was slightly less than two seconds (still relatively high).

After several code reviews and tests, we identified the problem. To reduce noise, the code was performing an average of the last 20 values of distance and using this averaged value to dictate whether the motor should vibrate or not. After placing an obstacle in front of the sensors, a new distance value was being added to the old 19 values, producing a new mean. However, when the older 19 values were somewhat different from this new distance the new mean was not so distant to the old mean. Therefore, the delay was happening because it was taking a longer time for the averaged value to get closer to the real distance.

The solution was simply to ignore the average, and leave the instantaneous distance values from the sensors dictate the vibration. The result was very satisfactory and the LV-MaxSonar sensors didn't present a severe problem regarding noise. The moments where the motors vibrated without obstacles near were rare.

**Decision and Important Note.** As presented above, several factors contributed to the final decision of using the LV-MaxSonar model in the final prototype. The most relevant one was definitely the accuracy when identifying people. It wasn't practicable to use the PING))) sensor in the project. The main idea of this concept was to detect obstacles so that the user don't collide with them while walking. Therefore, using a sensor that for several periods of time simply miss soft objects wasn't a feasible possibility. The LV-MaxSonar model presented excellent results with clothes and, with the delay problem solved, was a solid choice for the new prototype.

***Note.*** All the tests described between subsections "Previous Prototype" through "Sleep Function and Optimization in Energy Consumption" were performed with the PING))) sensors attached to the system. To also be able to justify the same conclusions and decisions that arose from those tests for the LV-MaxSonar EZ1 model, some other measurements were performed to ensure the power-consumption similarity between sensors.

Table 5 presents the measured current leaving the PowerBoost for the exactly same situations as the PING))) sensor was submitted, when only the battery was providing the energy. The values from the PING))) sensor were added for comparison.

The values show that, not only the proposed new sensor is similar to the previous one, validating all the former conclusions, but also it requires less energy from the battery.

Table 5

*Current leaving PowerBoost. Both sensors comparison (Only Battery)*

| | Current Leaving PowerBoost | |
|---|---|---|
| | LV-MaxSonar EZ1 | PING))) |
| Button Off (Sleep On) | 7 mA | 7 mA |
| Button On, No Obstacles | 40 mA | 69 mA |
| Button On, Closest Obstacle in all three | 160 mA | 170 mA |

### Vibration Motors' Output Mode

We changed the code to support different vibration intensities, and then tested the prototype with these changes. However, the difference between two vibration intensities was barely noticeable. Therefore, we thought that the intensity output could not serve the purpose of indicating distance. We discarded the changes and went back to telling distance from the length of the interval between vibrations.

## Solar BatPACK

Solar BatPACK stands for Solar Battery Powered Awareness of Closeness Kit. Considering all the tests and discussions mentioned on the sections above, we set the new configuration for this new prototype.

### Components and Circuit

The circuit was assembled as Figure 3 shows. The list of components used in this project can be seen in Table 6. The table also refers to each component as the correspondent letter presented in Figure 3.

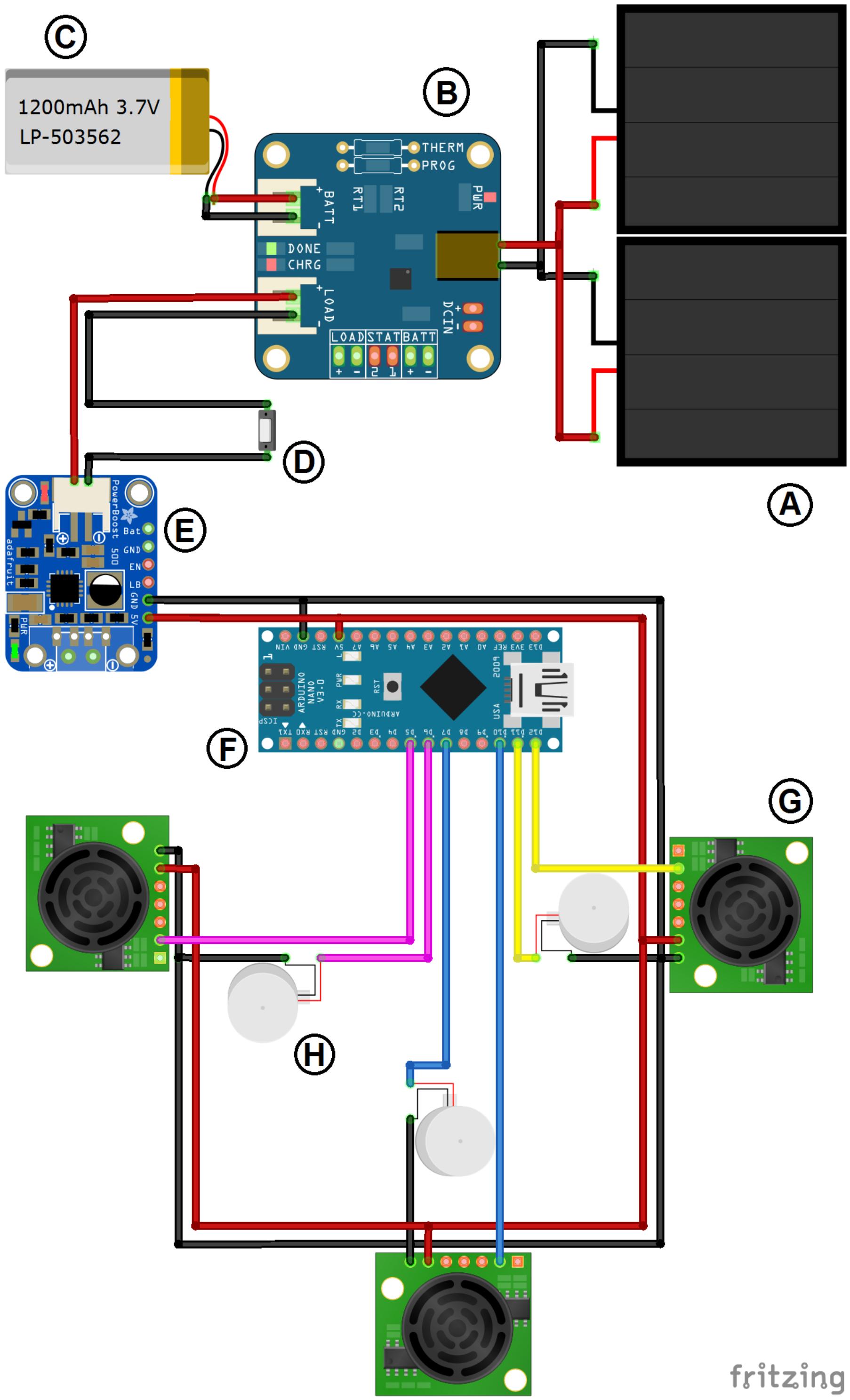


*Figure 3*. Solar BatPACK Circuit Diagram.

Table 6

*List of Components used in the new Prototype. Letters refering to Figure 3*

| Letter Reference | Quantity | Type | Model Name |
|---|---|---|---|
| A | 2x | Flexible Solar Panels | AdaFruit 6V 1W |
| B | 1x | Charger | USB/DC/Solar Lipo Charger |
| C | 1x | Lipo Battery | LP-503562, 3.7V 1200mAh |
| D | 1x | Slide Switch | Breadboard-Friendly SPDT |
| E | 1x | PowerBoost | 500 Basic - 5V USB Boost |
| F | 1x | Arduino | Micro |
| G | 3x | Ultrasonic Sensor | LV-MaxSonar-EZ1 |
| H | 3x | Vibrating Motors | AdaFruit's Mini Motor Disc |

Without any sun, or inside closed spaces, the battery provides all the power to the Charger. When the switch is off, no energy can go to the PowerBoost and there is a negligible amount of energy being wasted (28.3 $\mu$A leaving the battery, as mentioned). With the switch on, the current flows to the PowerBoost which provides the correct voltage to the system (5V).

The PowerBoost's 5V pin is connected with the Arduino's and with the three sensors' 5V input, while its ground pin is connected with the Arduino's, the three sensors', and the three motors' ground input.

The sensors' third connection was made between their PW pin and the digital pins from the Arduino. The distance could be calculated using the scale factor of 147 $\mu$s per inch provided in the data-sheet.

The three motors' second connection was made between their red wire and the digital pins from the Arduino. When these digital pins were set to HIGH in the Arduino, the current went to the motors, causing them to vibrate.

With some insolation, the only difference in the system is that, with the switch on, the energy produced in the solar panels can help the system by decreasing the energy required from the battery. If the insolation is high enough, it can even charge the battery while on (as presented in the previous tests). With the switch off, any amount of energy produced by the panels charges the battery.

## Code

One modification we made in the code was the change in vibration levels. The other group used seven different vibration levels to represent distance. We noticed that the difference between two vibration levels was not very noticeable, so we changed that to four different frequencies. The frequency level for a distance can be calculated by dividing the distance value by 50, since the range is 200 cm.

There are two built-in functions in the Arduino code, *setup()* and *loop()*. In the *setup()*, we define the pins and initialize a timer object. The *loop()* function is the body of the program. We have a variable $i$ that changes between the three sensors every iteration. In the loop function, first, we call the *pingDistance(i)* function, with the current $i$ as a parameter. The *pingDistance()* function will update the value read by sensor $i$, and update the distance level for that sensor, which is distance/50. Afterwards, we call *timer.run()*. The method *timer.run()* will call *vibrateAll()* every 60 ms. The method *vibrateAll()* will iterate through the three motors and call *vibrate()* for each of them. In *vibrate()*, we check if the current distance of the object is inside the spectrum of vibration ($<$ 200 cm). If that is true, we vibrate according to the distance level, which means that we vibrate once and don't vibrate *distance level - 1* times. If the distance is greater than 200 cm, we do not vibrate the motor.

The figure below presents the code's diagram.

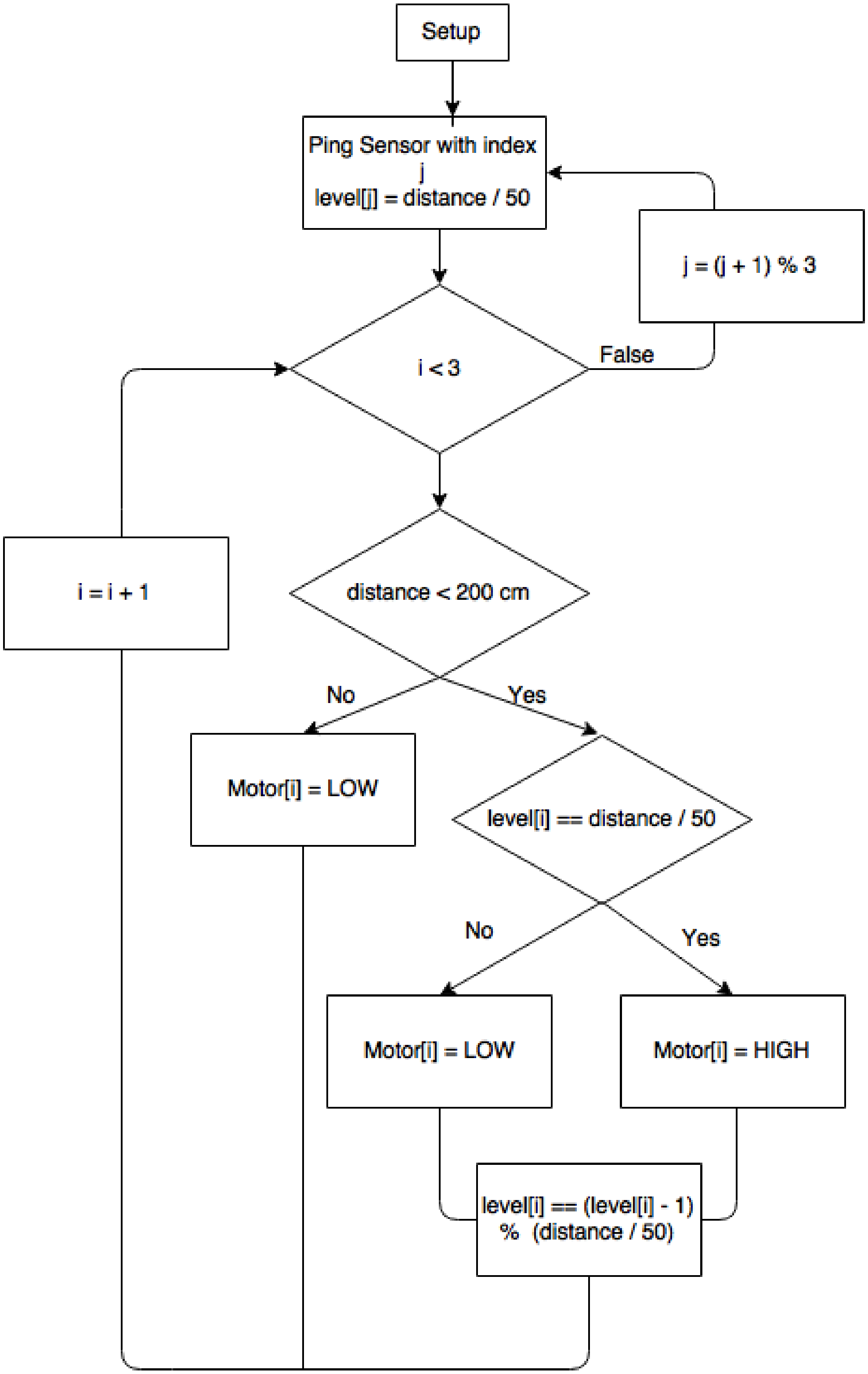


*Figure 4*. Code's Diagram.

## Conclusion

After all this effort we could conclude that there are several aspects that may affect the performance of the system. First it was needed to select the right sensors, since that each sensor has some advantages and disadvantages. The connections are also an important issue that must be addressed when building this kind of prototype. Bad connections may create noise and compromise the system. After a lot of work, the code is optimized when compared with the previous prototype and different problems were solved. After the changes on the solar panel and on how the system turns off, it is safe to say that the prototype can be used during a whole day without the need to be connected to the grid. Other aspects were chosen thinking about the user and the best way to implement this kind of system. That is why we decided to use a backpack and change the number of vibration levels.

In addition to that there are other problems that were considered but due to the lack of time weren't fully addressed. However, the group already thought of some solutions. This includes a pair of bracelets with extra vibrations motors that would be used during cold days when the clothes may interfere on the user's sensitivity. Also we thought about crafting a protection case necessary for some parts of the system. With all that been said, the BatPack works on different situations, has a great response time and is already fully functional. Some minor aspects need to be addressed, such as a more exquisite design.

## References

Chio, Y; Kim, Y; Oh, C; Okhandiar, N. & Illinois Institute of Technology. (2015). *Clothing Based Proximity Sensors for the Visually Impaired.* Retrieved from: https://github.com/jhajek/ITMT492-SPRING2015/tree/master/Clothing-Based-Proximity-Sensors-for-the-Visually-Impaired/Final_Embeded

Dubilier Industries. *Specification of Li-polymer Rechargeable Battery.* Retrieved from: http://datasheet.octopart.com/LP-503562-IS-3-BAK-Industries-datasheet-12495847.pdf

MaxBotix INC. *LV-MaxSonar-EZ Series High Performance Sonar Range Finder.* Retrieved from: http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf

Parallax INC. *PING))) Ultrasonic Distance Sensor Data-Sheet.* Retrieved from: https://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf