Team 4: Electric Juice

Final Report and User Manual

Date: 4/29/2000

Team Members:

Zachary Hoover zdh44@nau.edu

Zachary Nakama zkn2@nau.edu

Ryan Sasaki rss272@nau.edu

Table of Contents

1. Introduction

W. L. Gore & Associates is a company based in America that specializes in effective materials that can be useful in different applications. Gore was founded by Bill and Vieve Gore in 1958 and are best known for their innovative material called polytetrafluoroethylene more commonly known as GORE-TEX. Since then, Gore has been able to create derivative products from GORE-TEX fabric in different areas such as electronic components, medical devices, pharmaceutical & biopharmaceuticals, and many more.

Gore has been participating in the capstone program at Northern Arizona University to give students a taste of the type of projects they focus on. Our team was fortunate enough to be able to receive one of their project proposals and was given the opportunity to work with one of Gore's employees, DJ Mongeau. The project proposal that our team was given is titled Bioelectric Generator. For this project, we are to design, build, and test a fully implantable electric generating/charging system designed to recharge pacemaker-type implants to prevent the need for surgical battery replacement. On top of being biocompatible, the generating/charging systems must be non-inductive.

A pacemaker is a quarter-sized device that is implanted right under the skin of the left side of the chest. Extending from the pacemaker itself are two leads that are inserted into the heart. The device is designed for patients with arrhythmia which is a heart disorder where the patient's heart is unable to function correctly which may be fatal. The pacemaker's role is to send electrical impulses, which shock the heart back into rhythm. This stabilizes the patient's heart arrhythmia and allows their heart to pump the blood needed for them to function.

Pacemakers are battery-powered devices. Due to this, there are some problems that arise. First, the device's battery lasts about 5-15 years, varying depending on the amount of use a patient needs. For a disorder that lasts until death, there will be numerous surgical battery replacements throughout one's lifetime. These battery replacements can vary from $20,000 to $95,000. This is a very hefty price for a procedure that needs to be done every 5-15 years. On top of expensive battery replacements, the patient involved in this process is prone to complications after surgery such as infections at the site of the surgery or battery failure.

The problems mentioned are the reasons why the pacemaker needs to be wirelessly rechargeable. Since W. L. Gore & Associates is a company that specializes in materials in the medical field, they assigned us the task to solve these problems. Gore will be sponsoring our team by providing $3,000 of funding to cover our documentation, materials for testing and prototyping, and construction of multiple models that can help advance existing research in creating an effective and affordable design for a pacemaker generator.

2. Design Process

The design process of this project consisted of two major parts which were research and prototyping. During the research process our team studied different prototype designs that were either already created or had significant research done on them. The four major possible solutions to our problem that we found useful talked about piezoelectric generation, photovoltaic cell array generation, triboelectric nanogenerators, and pulsatile generation. The main goal for the research process was to find possible solutions to our main problem of generating electrical energy for a pacemaker battery. Out of these four different possible solutions we needed to choose one that would fulfill our needs best. Our team first ruled out triboelectric nanogenerators and pulsatile generation because of the mechanical complexities and the increased patient health risk. The other two methods, piezoelectric generation and near-infrared (NIR) light generation, were what we believed were the most feasible. This brought us to the second part of our design process, prototyping.

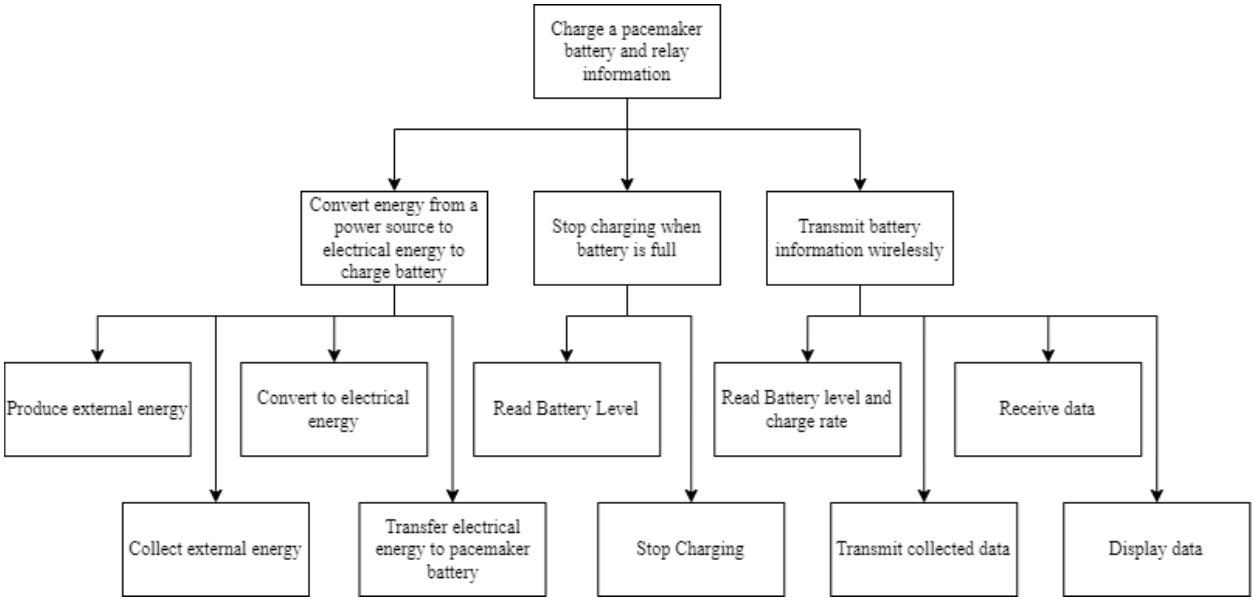


Figure 1: Functional Decomposition

2.1 Prototypes

After finishing our initial research, our next step was prototyping. As a team, we decided that our most important task was to prototype the charging methods we believed were most feasible, which were NIR light generation and piezoelectric generation. The third prototype focused on our stretch goal, the wireless communication system. We chose that as our third prototype because we believed that our battery charge management system was something that we could implement with less research and time than the stretch goal. With this, each team

member began working on an individual prototype, these being photodiode generation, piezoelectric generation, and the wireless communication system.

2.1.1 Photodiode Generation

Within our functional decomposition, *Figure 1*, this aspect of the prototype would assume the role of converting energy from a power source to electrical energy to charge a battery with the power source being NIR light. Before starting this prototype, our team expected to learn more about the charging process of LiPo (Lithium-ion Polymer) batteries, as well as how photodiodes, phototransistors, and photoresistors worked. We expected our biggest challenge would be learning how to choose which photodiode, phototransistor, or photoresistor we would want to use for our project. We knew that the end goal would need the prototype to be responsive to NIR light, but it took some time and research to find the sensitivities of the various power generators. Since our team didn't know the final charging method we would choose to power a pacemaker, this prototype was integral in our realization of a feasible way to generate power within the human body.

Our team performed a successful prototype demonstration on Thursday, November 18th. While demonstrating, we showed the circuit, which included a ground wire connected to the cathode end of a BPW34 photodiode. On the anode end, a 100 ohm resistor was connected to the A1 pin on an Arduino Uno board. The Arduino was connected to a computer. The A1 pin reads the voltage coming from the resistor connected to the photodiode, which was used to calculate the current. The demonstration results went as expected, as we had tested thoroughly during the nights before.

The largest challenge we had with this prototype was properly gathering data on the components we planned to use to make the prototype. It took a large amount of research to decide on which photodiode, phototransistor, or photoresistor we wanted to use. We chose the BPW34 Photodiode for the prototype after considering the reverse light current vs. the illuminance and relative spectral sensitivity vs. wavelength graphs of various components; the graphs are shown in Appendix A. The time needed to complete the prototype was a few hours longer than expected, as there was a need to refresh ourselves with infrequently used knowledge. This included Arduino programming and some basic circuit elements and functions. If we could redo the prototype, we would also try integrating some sort of battery charge management system. While doing research after completing the prototype, we came across an IC (Integrated Circuit) that could perform many of the duties needed for the battery charge management system. Using this in conjunction with the photodiode prototype success benefited the overall project's progress greatly. The results from this prototype were a deciding factor in how we will go about power generation in our overall project.

2.1.2 Piezoelectric Generation

Piezoelectric Generation, turning mechanical energy into electrical energy, was our second potential method of power generation for the pacemaker battery. This prototype consisted of six different electrical components, these being a 100 ohm resistor, a 1μF capacitor, a piezoelectric wafer, a red LED (light-emitting diode), and wires to connect each of them. What we expected to learn with this prototype was to determine if using a piezoelectric generator would be a possible solution to charging a pacemaker. Similar to the previous prototype, we expected some challenges in circuit design. This prototype reduced risk to the project since experimenting with different energy generating methods allowed our team to eventually decide on one method that will work best with our situation.

On the day of the demonstration, our team was able to partially demonstrate the circuit that we had created. During the demonstration, the circuit was constructed incorrectly because we had the button before the capacitor, resulting in the capacitor not being charged. Instead, we demonstrated the LED being activated when directly connected to the piezoelectric wafer. Our intent was that the piezoelectric wafer would charge the capacitor, and after it was charged, the button would be pressed to light the LED. After quickly rewiring the circuit, the results of the demonstration were what we expected.

The biggest challenge to making the prototype functional was the circuitry. One of the problems was incorrectly connecting an LED, which has an anode leg and cathode leg (positive and negative respectively). When connected incorrectly, theLED blocks current and does not produce light. We initially connected it incorrectly, leading to the LED not lighting. Another issue we had was that we connected the button before the capacitor which led to the capacitor not being charged when we wanted. We fixed this and were able to rewire the circuit to make the piezoelectric wafer charge the capacitor and release the electricity when the button was pressed. If we could reattempt this prototype, we would test if it could charge a battery. Since we were only able to light an LED, we still did not know how the circuitry would change if it were charging a battery. The results from this prototype influenced the project heavily. While trying to charge the capacitor with the wafer, we found that the amount of force needed to get 0.5 volts (V) is more than we expected. What we believed would be slight taps on the wafer was actually similar to setting a nail. Due to the nature of this device's implantation, we realized that it would be difficult for the patient to try and tap something located inside of their body. We concluded that this charging method was not feasible.

2.1.3 Wireless Communication System

Within our functional decomposition, *Figure 1*, this prototype would assume the role of transmitting battery information wirelessly. We expected to learn how to wirelessly transmit the data we desired by completing this prototype. Our team believed the most challenging part would be getting the Arduino and Raspberry Pi to communicate with each other via Bluetooth.

This prototype gave us some headway on our stretch goal of wireless communication of battery information.

During the demonstration, the Arduino properly transmitted data wirelessly to a smartphone via a Bluetooth signal. This result was a partial success as the Arduino was able to wirelessly transmit data to an outside device, but the Raspberry Pi was not able to receive data as we did not have the proper code ready for demonstration by the deadline. The results were as we expected, since the demonstration was relatively simple: it consisted of uploading a small amount of code to the Arduino and then using a smartphone to receive the information.
The biggest challenge in making the prototype functional was learning what was needed in order to properly transmit data using an Arduino's internal Bluetooth. We also had very little experience with Raspberry Pi's, so we were starting from scratch when it came to learning what was necessary to set up any form of Bluetooth communication. The prototype took more time than expected, and resulted in not being able to implement the Raspberry Pi completely by the deadline. If we were able to do this prototype again, we would have researched the basic functionality of a Raspberry Pi prior to prototyping and started earlier to complete everything expected. The results of this prototype influenced the project heavily by showing us what we need to focus on with respect to the wireless communication system.

2.2 Prototype Conclusions

During testing of each prototype, we saw the flaws of each possible solution and determined that NIR photodiode generation had the least amount of flaws and solved the problem the best. It had more feasibility compared to piezoelectric generation as we could not brainstorm a way to sufficiently provide the force needed to create enough power without providing further harm to the patient. It was also easier to properly test as we did not have a way to repeatedly receive consistent voltage from the piezoelectric wafer. Our partial success with the wireless communication prototype also led us to further pursue Bluetooth as our means of transmitting data wirelessly.

3. Final Design

Our system architecture, shown in *Figure 2*, visually explains our team's final design. Our final design is separated into two blocks: the components within the body and the components outside the body represented by blue and red, respectively. The components in the blue box are contained within a small case. In the blue box, the BPW34 photodiode array is connected to the L6924D IC, our battery charge management system, which regulates the power coming from the array. The photodiode array is also powering the Arduino Nano 33 BLE. The Arduino Nano 33 BLE reads the battery level and charge rate from the charge management system, which then transmits the data to the components outside the body. The charge management system transfers the regulated power to the pacemaker battery. The Raspberry PI receives the data the Arduino Nano 33 BLE transmits via Bluetooth. The Raspberry Pi then shows the received data on a display, letting the patient know the current state of the battery. The final component in the red box is the Raspberry Pi power supply, which connects the Raspberry Pi to a typical American wall outlet. In the rest of this section, we will describe the major components used: the BPW34 photodiode array, the L6924D IC and PCB, the Arduino Nano 33 BLE, the case that will contain previous three components, the Raspberry Pi 3 Model A+, and the display.

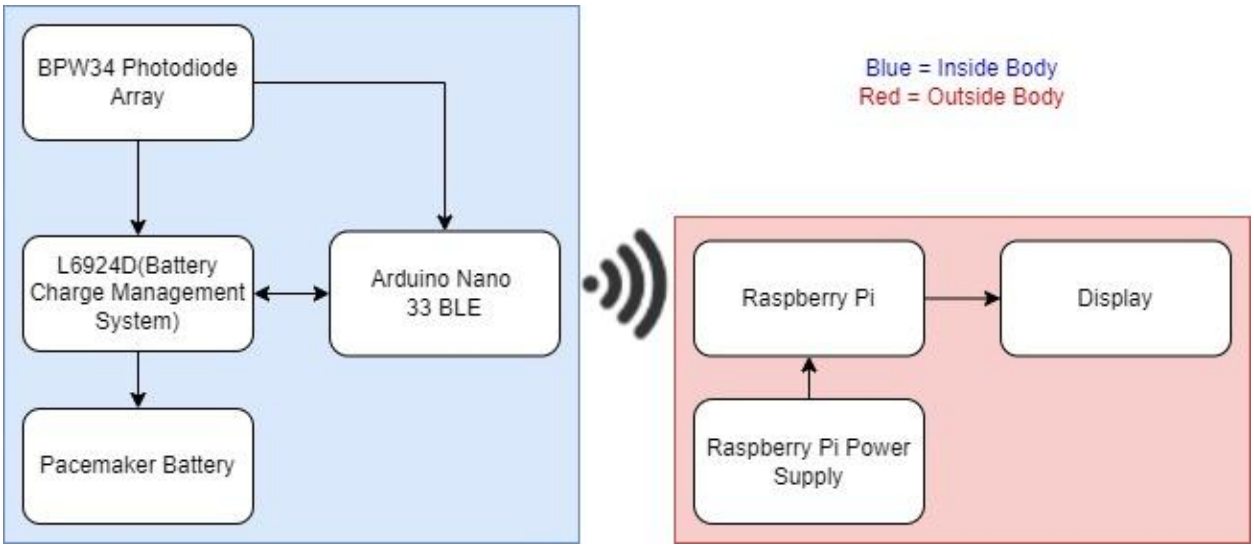


Figure 2: System Architecture

Shown in *Figure 3*, our first component is the BPW34 photodiode array. This photodiode array collects NIR light, our chosen energy source. The array consists of 100 photodiodes soldered onto a PCB (printed circuit board). They are arranged in 5 groups of 20 photodiodes in series, with each group of 20 connected in parallel. Each individual photodiode measures 5.4 x

4.3 x 3.2 mm (L x W x H), as seen in Appendix B. The dimensions of the Capstone PCB v1.0 are 70 x 85 x 1 mm.

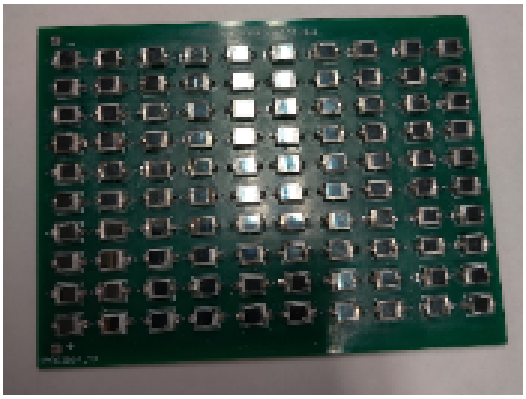


Figure 3: BPW34 Photodiode Array on Capstone PCB v1.0

Our second component, shown in *Figure 4*, shows our completed L6924D PCB. This charge management system consists of a PCB with resistors, capacitors and an IC soldered on. The resistors and capacitors are soldered around the IC, adjusting the charging current, the end-of-charge current, and output voltage (equations used to calculate values in Appendix C). Tasks like stopping charge from flowing when the battery power is full or start charging when the battery is dead is done through the charge management system. The dimensions of the charge management system PCB are 35 x 45 x 1 mm. Values used for resistors and capacitors are in Appendix D.

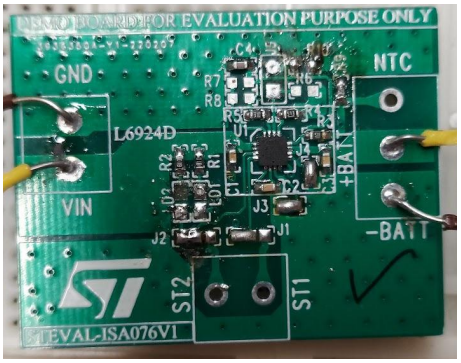


Figure 4: L6924D Charge Management System

Our third component is the Arduino Nano 33 BLE, shown in *Figure 5*. The Arduino was purchased pre-built, what we designed was the program to complete the tasks we needed. This program had two primary tasks. First, the Arduino reads the battery level and charge rate the L6924D IC produces. Then, the Arduino uses its built-in Bluetooth adapter to send the data to the external Raspberry Pi 3 Model A+. The complete Arduino program can be seen in Appendix E. The dimensions of the Arduino Nano 33 BLE are 45 x 18 x 1 mm.

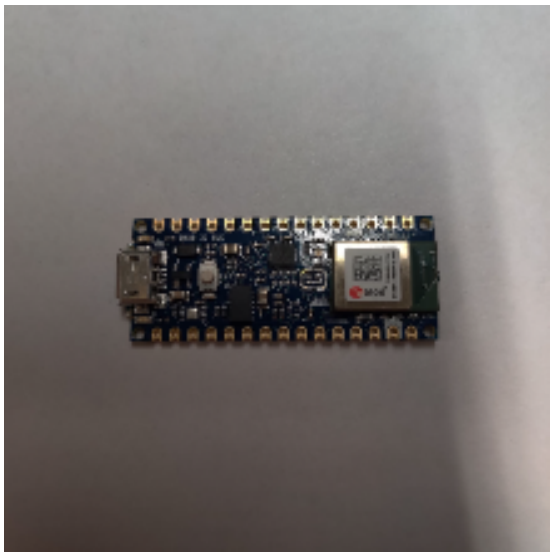


Figure 5: Arduino Nano 33 BLE

The case we used to contain the BPW34 photodiode array, the L6924D IC and PCB, and the Arduino Nano 33 BLE is shown in *Figure 6*. The case design is a simple rectangular prism 3D printed using thermoplastic PLA(Polylactic Acid). There are slots carved out to hold the Arduino Nano 33 BLE on the right and the charge management system on the left. A small track is carved out on the edges of the case to allow the PCB photodiode array to slide in as the lid of the case, as seen in *Figure 7*. This covers all the components within the case. To allow the wires to easily move around, holes were drilled in the end and in the cross bar inside the care. The dimensions of this case are 77 x 92 x 11 mm.

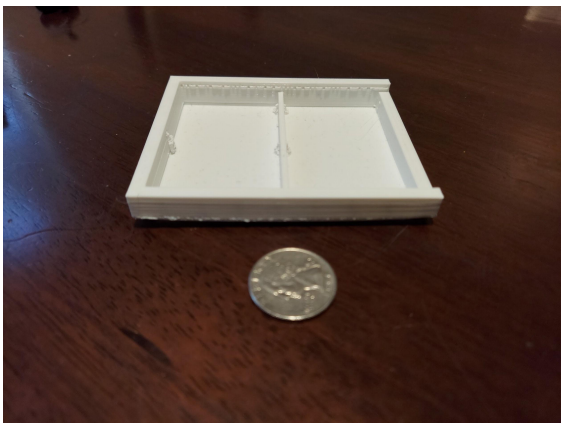


Figure 6: Final Case Design

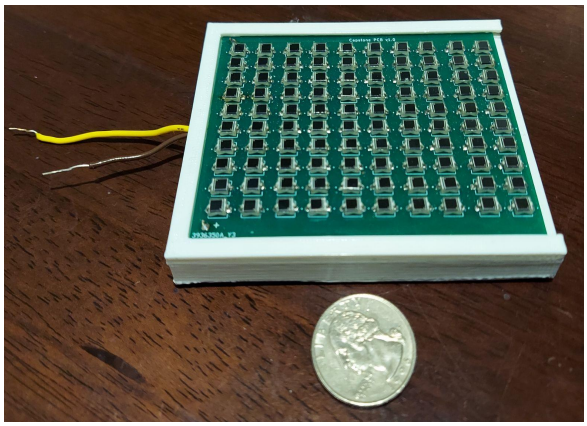


Figure 7: Case With PCB Array

Our first component located outside the body is the Raspberry Pi 3 Model A+, shown in *Figure 8*. This component was purchased pre-built like the Arduino Nano 33 BLE. The Pi

connects to the Arduino using a bluetooth signal in order to receive data about the battery. This data is then shown on the connected display. The Pi will also make you aware if the battery is fully charged or if there is an issue associated with the charging. The final measurements of our the Raspberry Pi was 6.5x5.6cm

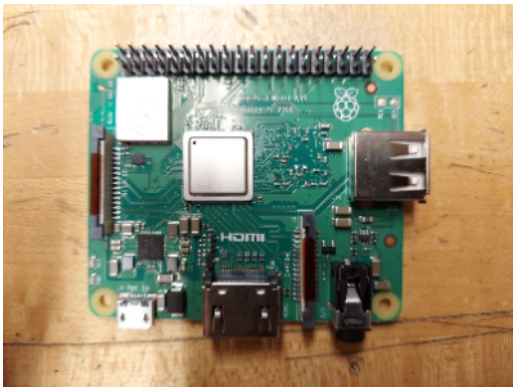


Figure 8: Raspberry Pi 3 Model A+

Connected to the Raspberry Pi 3 is the ElecLab Raspberry Pi Touchscreen Monitor, shown in *Figure 9*. This monitor displays the battery level and charge rate that the Arduino Nano 33 BLE reads and transmits to the Raspberry Pi 3. The raspberry pi is screwed onto the back of the display and connected using a DSI FSC cable**.**

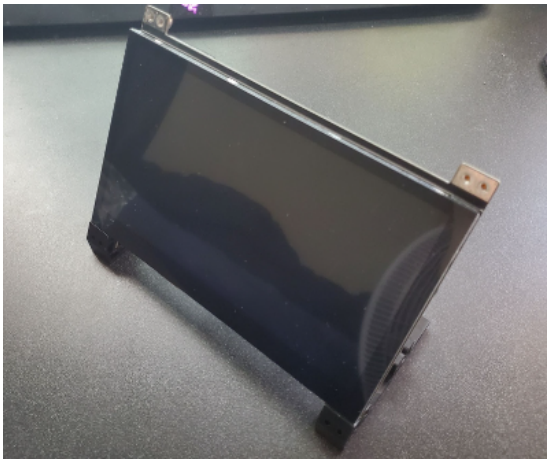


Figure 9: ElecLab Raspberry Pi Touchscreen Monitor

4. Results

4.1 Requirements

Type of Test	Status	Req #	Requirement
		1	**Size of implanted device**
Inspect		1.1	Size under 8cm x 8cm x 1.5cm
Inspect		1.1.1	Preferred under 6cm x 4.5cm x 1cm
		2	**Charge Requirements**
Inspect		2.1	Cannot use inductive charging (no electromagnetic or coupling)
Integrate		2.2	Does not damage battery by overcharging
UTS		2.3	The implant operates at a temperature below 95°F
UTS		2.3.1	Stretch: below 80°F
UTS		2.4	IC does not operate if a battery is not connnected
		3	**Materials**
		3.1	All materials are biocompatible
		3.1.1	Implanted material for shell do not produce an immune response over long periods of time (~10 years)
		3.1.2	Materials inside shell do not severely deteriorate over long periods of time (~10 years)
Inspect		3.2	All electrical components are completely contained within a "biocompatible" shell
Inspect		3.3	The shell of the implant does not interfere with charging method
		4	**Power output**
		~~4.1~~	~~Provide enough power to supplement heavy strain~~
UTM	*	4.1	Photodiode array outputs between 5-19mA
		~~4.2~~	~~Photodiode array outputs between 1-10V~~
UTM	*	4.2	Photodiode array outputs between 4-15V
		5	**Wireless Communication**
Inspect		5.1	Display current battery status. This includes:
UTS		5.1.1	Wireless transmission of battery charge level as a percentage to external display
UTS		5.1.2	Wireless transmission of battery charge rate to external display
		6	**Arduino**
Integrate	*	6.1	Disables IC when battery is at 3.5V
UTS		6.2	Arduino can transmit data via Bluetooth
UTS		6.3	Arduino can turn IC on and off

Figure 10: Requirements taken from Testing Workbook

Inspect = Inspection
UTM = Unit Test Matrix
UTS = Unit Test Step-by-Step
Integrate = Integration Test
Asterisk (*) = Major Requirement

4.2 Major Requirements

There are three requirements marked as major requirements. These are requirements:
4.1 The photodiode array outputs between 5-19mA,
4.2 The photodiode array outputs between 4-15V, and
6.2 The charge management IC is disabled when the battery is at 3.5V.

The first two requirements are considered the most important due to the nature of our task, creating a charging system. We could not create a sufficient charger without having enough voltage and current. Either the charger would not charge the battery fully or it would take an exceedingly long time to completely charge the battery. The third requirement is important due to hardware and user safety. If a LiPor battery overcharges, the battery could overheat, expand, and explode. Obviously, if our battery charger causes the battery to be damaged, then that is a catastrophic failure. Even worse, with the battery we aim to charge being located inside the body near the heart, the resulting damage would severely injure or kill the person using it.

4.3 Major Tests

There are 3 major tests we performed that correspond to our major requirements. These tests are the UTM Photodiode Array PCB Current test, the UTM Photodiode Array PCB Voltage test, and the UTS Arduino Functionality test. UTM is used when the inputs are structurally the same and differ only in value. UTS is completed by creating step-by-step instructions for generating a test and checking the results. The instructions should be able to be clearly replicated. We believe that these tests are the most important due to the first two being necessary to charge a battery, the goal of this capstone, and the third being a necessary safety precaution to prevent the accidental overcharge of the battery.

The first major test that our team conducted is the UTM Photodiode Array PCB Current test for the requirement 4.1-Photodiode array outputs between 5-19mA. Here, we needed to verify that our array of 100 photodiodes was able to output a current of at minimum 5-19mA. This value for current was crucial for our final prototype because it needed this amount of current to perform the task of charging the battery at a reasonable speed. The process of verifying that our photodiode array met this requirement required two things, a tape measure/ruler and a multimeter. The process of setting this test up and completing starts with attaching a multimeter to the photodiode array to properly measure current. Then, place the NIR lamp 2 inches away and power on the lamp and multimeter. Then, using the multimeter, record the current produced. Finally, increase distance by 2 inches each step until you have reached a distance of 12 inches. At each step we recorded the measurements from the multimeter and documented the data. Below, in *Figure 11*, is the plot that our team generated from the data points that we collected. We can see that the benchtop

current was much lower than the actual current produced by the photodiode array, but eventually met up with the current that we expected the further we placed the near infrared lamp. The benchtop current was based on our previous testing with the photodiode array assembled on a breadboard.

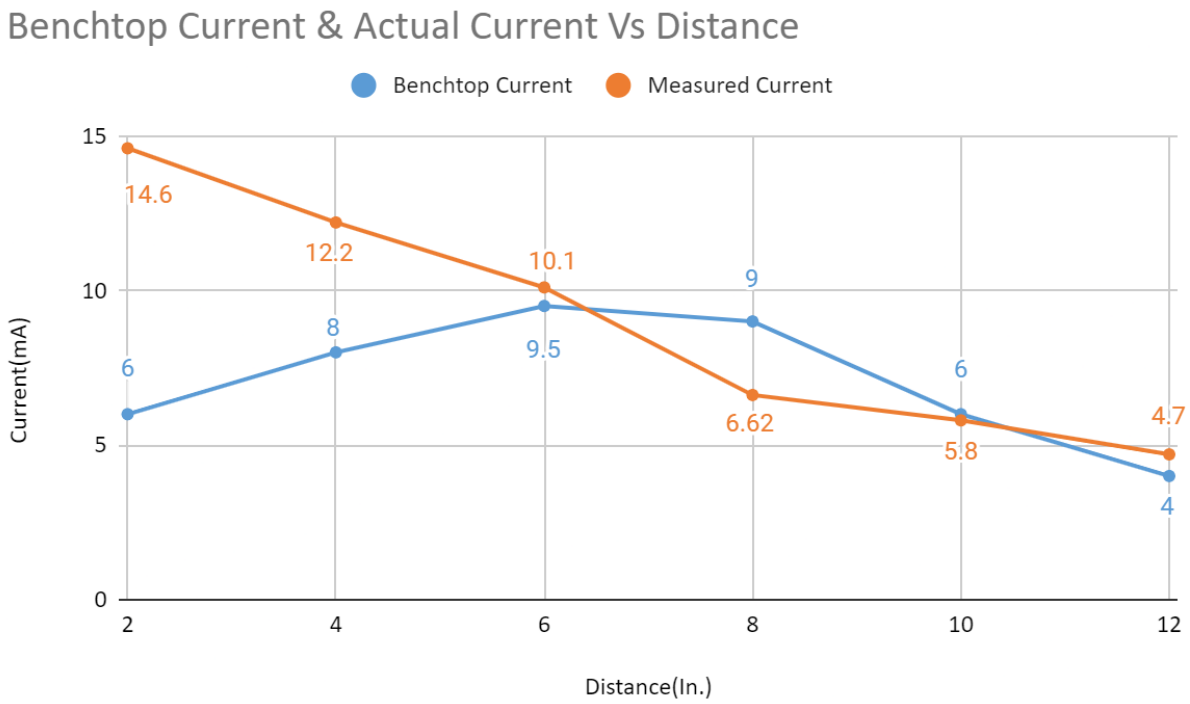


Figure 11: Photodiode Array PCB Current vs. Distance

The second major test we performed was the Photodiode Array PCB Voltage test. This test was based on requirement 4.2: Photodiode array outputs between 4-15V. To properly execute this test, we needed a measuring tool and a multimeter. With the measuring tool, the NIR lamp is placed 2 inches away from the photodiode array. The multimeter is connected to the terminals of the photodiode array. With the multimeter turned on to read DC voltage, the NIR lamp can be activated. The data is recorded and the test is repeated, increasing the distance by two inches each time until a final distance of 12 inches is reached. With this data, we can now calculate the voltage the photodiode array can produce at any distance within 12 inches. Our expected voltage was lower than the actual voltage in both the short and long distances, but it exceeded the actual with the middle distances. Below in *Figure 12*, this can be seen in the line graph comparing voltage versus distance.

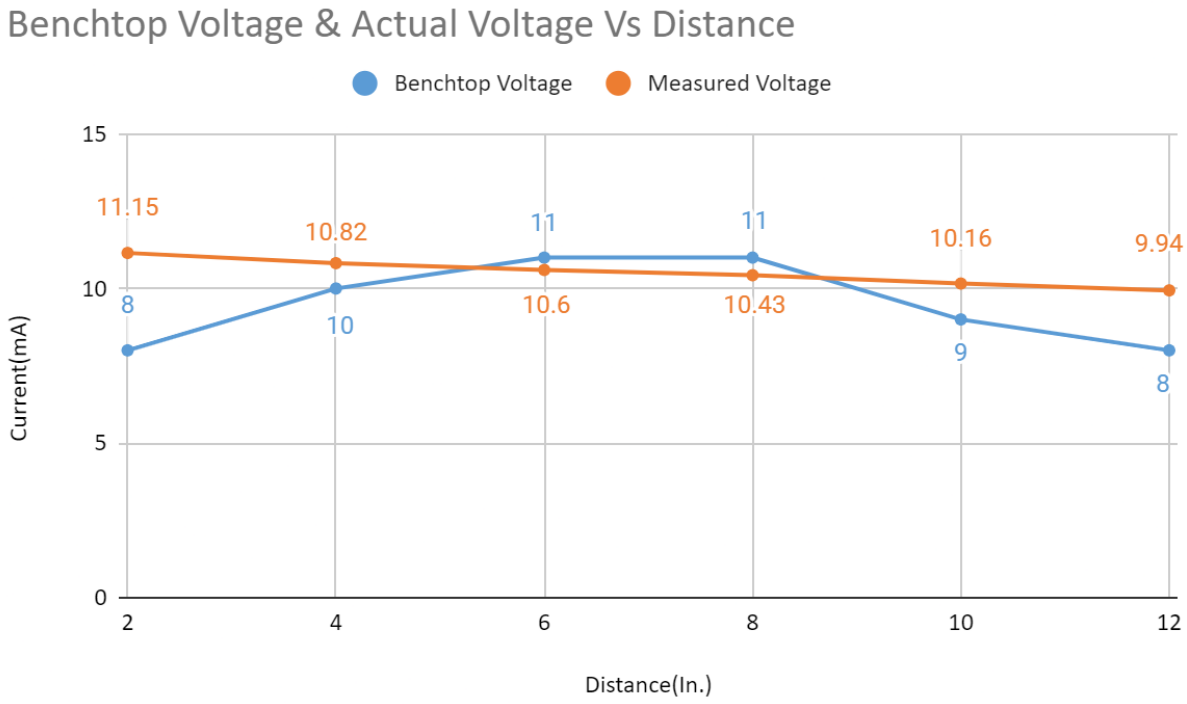


Figure 12: Photodiode Array PCB Voltage vs. Distance

The last major test that our team conducted was the UTS Arduino Functionality test for the requirement 6.3-Arduino can turn IC on and off and 2.2-Does not damage battery by overcharging. With this test, we needed to verify that our Arduino can properly shutdown our IC when the battery is fully charged. This characteristic plays a very important role for our whole prototype because if it cannot stop charging it would eventually put stress on the battery and damage it. To set up the test, we first needed 4 things to run the test: a micro-USB to USB-A cable, a laptop, a DC power supply, and a multimeter. After obtaining the needed supplies we connected the Arduino Nano 33 BLE to the laptop for power. Then, we connected the pins D2 to the multimeter, pin A5 to the DC power supply, and pin GND to ground. The DC power supply will mimic our battery in this test. We can then change the DC power supply to 2V, a voltage below the max capacity of the battery, and measure the voltage coming out of D2. This will verify that when pin A5 is at 2V the voltage coming out of D2 should read 0V. When we increase the voltage of the DC power supply to 3.5V, the pin A5 is at a voltage of 3.5V which leads to D2 producing

approximately 3.3V. In *Figure 13* below, we can see that when the SD/A5 pin is at a voltage below 0.4V , the IC is enabled; at a voltage of 2V and above, the IC is disabled. This gave us the results that we predicted we would receive from the test. In Appendix E, the Arduino code that performs this process is shown.

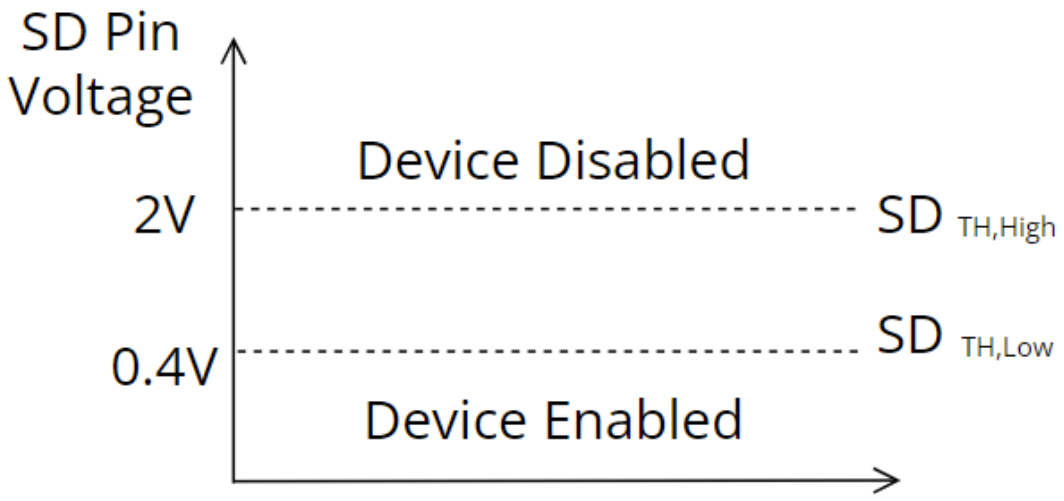


Figure 13: SD pin shutdown

4.4 Analysis of Results

When provided with a completely depleted battery, the NIR photodiode biolectric generator successfully charged the battery to an acceptable voltage of over 3V in approximately 20 minutes. These results were as expected and are within what was designated as a successful charging. The most important requirements related to the functionality of the device are the successful charging of the device and the safety of the user. Both of these requirements were met during testing. The stretch goal requirement of having wireless communication implemented into the design was only able to be half achieved as data is able to be sent, however the external display is unable to properly receive the data.

5. Conclusion

The pacemaker device is an incredible technology that revolutionized the treatment for heart arrhythmia, but the danger of a failing battery, the risk of pacemaker replacement surgery, and the replacement cost reduce the unquestionable benefit of having a pacemaker device. Our team was able to successfully design and test our bioelectric generator that can fully counter the drawbacks of the non-rechargeable pacemaker. The device that our team created meets our three major requirements that our team set. Our device successfully charged a 3.7V battery from 0V to 3V in 20 minutes in a safe manner for both the client and the battery, the charging would stop when the battery was near capacity, and the device charged the battery using NIR light. Considering that a normal pacemaker battery lasts 5-15 years, the results that we achieved are excellent as the patient would only need to use our device for about 20 minutes whenever the battery is low, as opposed to replacement every 5-15 years.

Although our team considered this device an overall success there were still some aspects that we fell short of. We partially completed the wireless communication system. The Arduino Nano 33 BLE could properly read the battery level and charge rate and then transmit that data wirelessly via Bluetooth. Our issue occurred when trying to receive that data on the Raspberry Pi 3 Model A+. The Raspberry Pi would successfully connect to the Arduino and receive the data, but only for the first ping. After the first ping, the Raspberry Pi would disconnect and would need to be manually reconnected to receive further data. Our goal was to have the Raspberry Pi constantly receive data but as this could not be done, we consider the wireless communication system to be a partial success.

Another aspect we could improve upon is the total size. The size of the device was bigger than it could have been due to inexperience in PCB design. With the experience we gained, we could decrease the distance between the photodiodes even more. A possible smaller design can be seen in *Figure 14*. The Capstone PCB v2.0 would have the dimensions of 67 x 70 x 1 mm, approximately ¾ of the volume of the Capstone PCB v1.0. The current limiting factor of the case as a whole is the photodiode PCB, so decreasing it would allow us to design a smaller case as well.

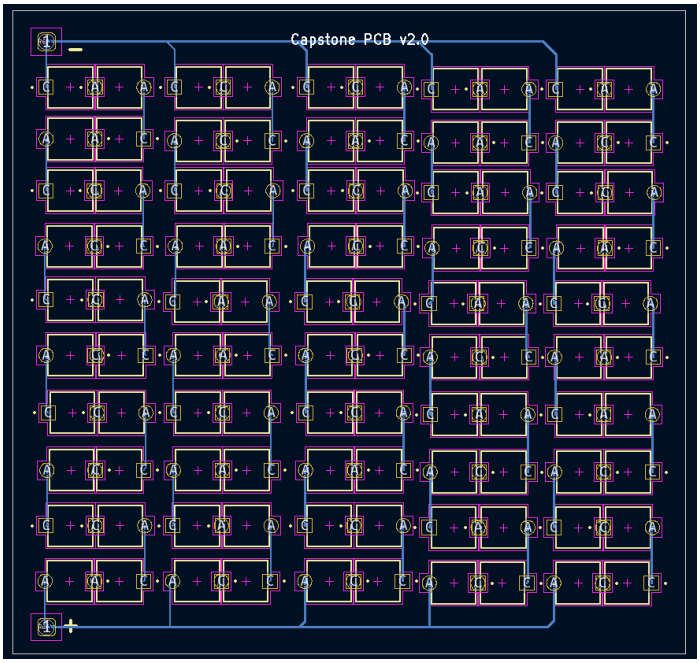


Figure 14: Capstone PCB v2.0

The power output of the photodiode array could also be improved. With the current photodiode array, both the voltage and current can be increased to have a faster charge time. An issue with this is that having more photodiodes in the array would lead to a larger array, and that contradicts the previous point. More testing would need to be done with smaller arrays with the current amount of photodiodes and larger arrays with more photodiodes to determine which design would fulfill our requirements best.

One of the biggest lessons we learned as a team is that there is never enough time to prototype and test. As we went through this semester and our project started to become functional, we realized that there are many improvements that could have been made if we started assembly earlier. This can be seen in the last two paragraphs where we saw improvements that we easily recognized with the experience we gained by completing an earlier version of the project.

Another lesson learned from this project is that the more research done beforehand, the less time will be spent on fixing errors made while physically assembling the project or writing the program. This lesson is obvious, but working on a project of a scale larger than any previous project gave us a new perspective on it. A few more hours of research could save a few days or even weeks of physical building time. Ordering a wrong component or misassembling a PCB because of a miscalculation are both things that can be easily avoided with a bit more research or studying of a data sheet.

6. User Manual

6.1 Introduction

We are pleased that you have chosen the Electric Juice pacemaker charging system for your business needs. Rechargeable pacemakers are a necessity, to remove the health risks and financial costs that come with a traditional non-rechargeable pacemaker. We developed this pacemaker charging system to eliminate the need for the invasive pacemaker battery replacement surgeries that come with traditional pacemakers. With our charging system, the number of surgeries performed is reduced, leading to patients being able to live a healthier life with a higher standard of living. The product that our team developed allows for pacemaker batteries to be charged wirelessly, via near-infrared light. Charging the pacemaker can be done by anyone, without the need of a scalpel, needle, or even medical experience (after the charging system has been implanted). Some of the key features include:

- Near-Infrared Light Powered: This charging system is based around the power of near-infrared light. Near-Infrared light is a type of non-visible light, due to the wavelength being around 850nm. Don't worry about it being harmful, near-infrared is similar to radio waves as they are both of longer wavelengths than visible light making both harmless to your body.
- Easy to view battery status: Using the built-in Bluetooth transmitter on an Arduino Nano 33 BLE, you can wirelessly view the charge rate and battery level. This Bluetooth signal typically goes to a Raspberry Pi 3 Model A+ to be displayed on a touchscreen, but any Bluetooth capable device can receive the data after being properly connected.
- Safety First: With the ease of charging the pacemaker battery with our charging system, what happens when the user inevitably falls asleep and forgets to stop charging the battery when it is full? Don't worry! Our charging system has a built-in charge cutoff thanks to the L6924D integrated chip. If you don't stop charging in time, the l6924D will stop for you. When the battery gets close to reaching capacity, the L6924D automatically brings charging to a halt.

The purpose of this user manual is to help you, the client, successfully use and maintain the pacemaker charging system in your life going forward. Our aim is to make sure that you are able to keep your heart healthy for many years to come!

The bioelectric generator is designed to lower the maintenance of the pacemaker device. The pacemaker's maintenance needs to be done every 5-15 years depending on the severity and frequency of the patient's heart arrhythmia. When the battery malfunctions or reaches the end of its life it requires a costly and risky procedure of replacing the pacemaker device. Replacement of the pacemaker device requires open heart surgery which puts the patient at risk of infection or medical malpractice. The bioelectric generator prevents the need for open heart surgery which in turn decreases the maintenance cost and provides a safer alternative to open heart surgery. For

our bioelectric generator to solve these main problems of a pacemaker battery we designed a device that contains three subsystems: a charging system, a charge management system, and wireless communication system. The charging system and charge management system is our solution to the degrading battery life of a pacemaker. The charging system harvests power from near-infrared light using an array of near-infrared photodiodes. The reason we chose near-infrared light was due to the safe and penetrative nature of near-infrared light. Near-infrared light is similar to radio waves, in that it can harmlessly move through you, but not to the degree of completely passing through as the wavelength of near-infrared light is not long enough. The charge management system allows the power generated from the near-infrared diodes to be regulated and transferred to the pacemaker battery safely without causing harm to the patient. The wireless communication subsystem mainly improves the convenience of our device, by allowing data to be collected and transmitted from the charge management system and displayed on an external screen to allow easy analysis of the battery's state as well as the current charge rate of the battery. Our final design of this project was deemed successful after fully testing and gathering data from the results. We were able to conclude that it was a functional, innovative solution to the project problem.

6.2 Installation

This section details how to set up the charging system. This system requires that the implant has already been surgically inserted into the body and successfully connected to the pacemaker battery. This device is recommended to be installed near a chair or bed for ease of use and comfortability of the patient, but it can be installed anywhere where the patient can keep the near-infrared (NIR) lamp steadily pointing at the implant.

1. Remove the NIR Lamp, stand, and Raspberry Pi Display from their respective packing, taking care to remove any additional plastic or packaging.
2. Attach the NIR Lamp to the included stand and adjust height to an appropriate level for the position of the implant on the patient.
3. Plug both the NIR Lamp and Raspberry Pi Display into a standard 120V AC wall socket.
4. Select and run the charging software on Raspberry Pi Display
5. Make final adjustments so that the lamp is at the recommended 4-6 inches away from the inserted device.

6.3 Configuration and Use

This section details how to operate the charging system once it is successfully installed. Once the system has been properly installed and orientated for safe use, the device must be connected to the display in order to display information, but not to charge the battery.

1. Power device and select and launch charging software.

2. If the display remains on the "Searching for Connection" screen for longer than 15 seconds, adjust the location of display in relation to the patient to be closer.
3. Once the display is in range of the device it should automatically establish a connection.
4. The display should now be displaying battery data including charge level and charge rate.
5. Turn on the lamp and remain still while the battery charges.
6. Once the battery is fully charged, the display will show this and the device will cease charging the battery.

6.4 Maintenance

In order to maximize the lifespan of the device all components should be kept in a dry environment and only operated for the duration required in order to completely charge the pacemaker battery. Should the supplied external display be damaged or broken, a replacement system may be acquired from Electric Juice or your medical provider and implemented without any additional changes or alteration.

6.5 Troubleshooting Operation

This section is intended to serve as a reference in the event that some aspect of the device is not operating properly. Should your issue not be discussed below, please contact the Electric Juice Team. Should your issue be related to your implant please contact a medical professional immediately or dial 911.

- The display will not connect to the implant or continually drop the connection.
    - This issue is most likely caused by the distance between the display and the implant. Move the display closer to the implant in order to ensure it is close enough to establish a connection. Should this not work, check if anything is obstructing the connection such as clothing or furniture. Make sure the display has an unobstructed path to the location of the implant.
- The display is connected but not updating.
    - While the display may not be updating, be aware the battery is still charging if the lamp is in position and turned on. The most likely solution to this issue is to restart the display by unplugging the wall outlet and plugging it back in. The display should reconnect to the implant and begin updating.

6.6 Conclusion

This concludes the user manual for the pacemaker charging system developed by team Electric Juice. Our team wishes the client and any users the best experience using our product. While we are all moving on to professional careers, we would be happy to answer short questions in the coming months to help you get the product deployed and operating optimally in your organization. We can be reached at zdh44@nau.com and will get back to you as soon as possible.

7. Appendices

Appendix A: BPW34 Reverse Light Current vs. Illuminance and Relative Spectral Sensitivity vs. Wavelength Graphs taken from BPW34 Datasheet

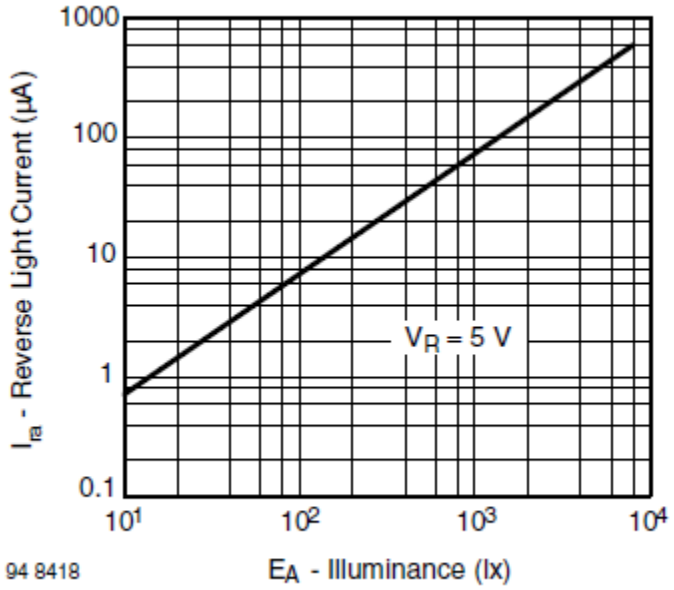


Fig. 4 - Reverse Light Current vs. Illuminance

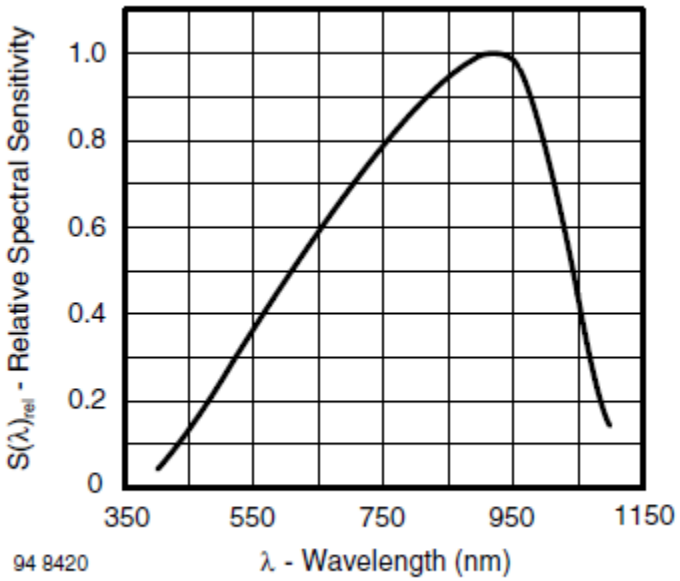
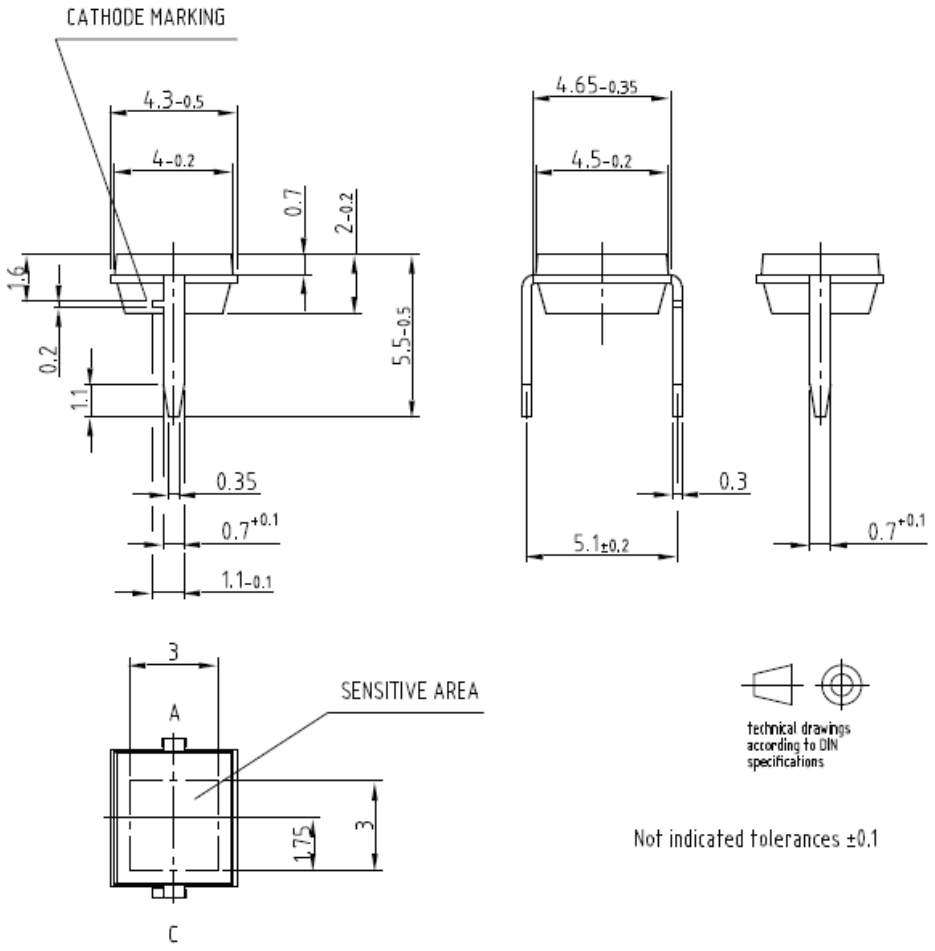


Fig. 7 - Relative Spectral Sensitivity vs. Wavelength

Appendix B: BPW34 Package Dimensions taken from BPW34 Datasheet

PACKAGE DIMENSIONS in millimeters



Appendix C: Charging Current, End-of-Charge Current, and Output Voltage Equations

Equation 9

$$R_{PRG} = V_{BG} \times \left(\frac{K_{PRG}}{I_{CHG}} \right)$$

I_{CHG} = Charging Current

Equation 10

$$R_{END} = V_{MIN} \times \left(\frac{K_{END}}{I_{ENDTH}} \right)$$

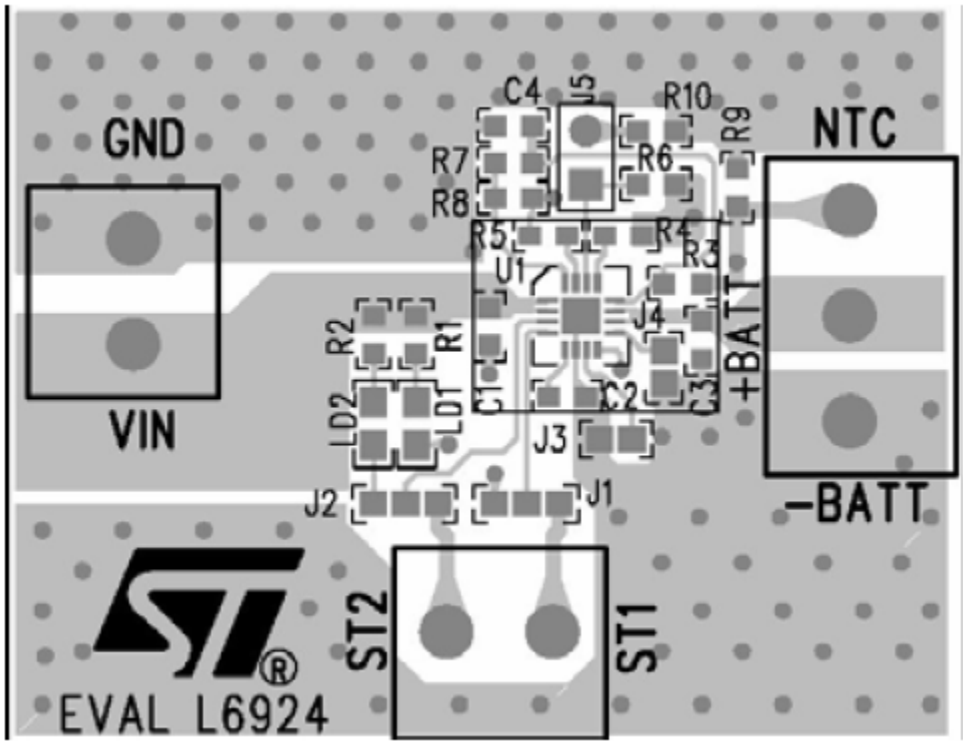
I_{ENDTH} = End-of-Charge Current

Equation 7

$$R_{PRE} = \frac{V_{REF} - V_{BG}}{\frac{V_{BG}}{R_{PRG}} - \frac{I_{PRECH}}{K_{PRE}}}$$

V_{BG} = Output Voltage

Appendix D: L6924D PCB Resistor and Capacitor Values



R1 = 1KΩ
R2 = 1KΩ
R3 = 510Ω
R4 = 51KΩ
R5 = 1MΩ
R6 = Open
R7 = Open
R8 = Open
R9 = 510Ω
R10 = Short
C1 = 1µF
C2 = 10nF
C3 = 1µF
C4 = 1µF

Appendix E: Arduino Nano 33 BLE Code

```
#include <ArduinoBLE.h>
BLEService batteryService("180F");
BLEUnsignedCharCharacteristic batteryLevelChar("2A19", BLERead |
BLENotify);
int oldBatteryLevel = 0;
long previousMillis = 0;

// CMS variables
int BatteryVoltageReadPin = A5;
int ControlPin = 2;
float BatteryCapacity = 3.5;
float BatteryVoltage;
float AdjustedBatteryVoltage;


void setup() {
  Serial.begin(9600);    // initialize serial communication
  while (!Serial);

  pinMode(LED_BUILTIN, OUTPUT); // initialize the built-in LED
pin to indicate when a central is connected

  // pinMode for CMS
  pinMode(BatteryVoltageReadPin, INPUT);
  pinMode(ControlPin,OUTPUT);

  if (!BLE.begin()) {
    Serial.println("starting BLE failed!");

    while (1);
  }

  BLE.setLocalName("BatteryMonitor");
  BLE.setAdvertisedService(batteryService); // add the service
UUID
  batteryService.addCharacteristic(batteryLevelChar); // add the
battery level characteristic
```

```
  BLE.addService(batteryService); // Add the battery service
  batteryLevelChar.writeValue(oldBatteryLevel); // set initial
value for this characteristic

  BLE.advertise();

  Serial.println("Bluetooth device active, waiting for
connections...");
}

void loop() {
  BLEDevice central = BLE.central();

  //calculate true battery voltage
  BatteryVoltage = analogRead(BatteryVoltageReadPin);
  AdjustedBatteryVoltage = BatteryVoltage * (BatteryCapacity /
1023.0);

  //if statements for whether battery is fully charged
  if (AdjustedBatteryVoltage >= BatteryCapacity)
  {
    digitalWrite(ControlPin,HIGH);
  }
  else
  {
    digitalWrite(ControlPin,LOW);
  }
  // end CMS loop

  if (central) {
    Serial.print("Connected to central: ");
    Serial.println(central.address());
    digitalWrite(LED_BUILTIN, HIGH);

    while (central.connected()) {
      long currentMillis = millis();
      if (currentMillis - previousMillis >= 1000) {
        previousMillis = currentMillis;
        updateBatteryLevel();
```

```
      }
    }

    digitalWrite(LED_BUILTIN, LOW);
    Serial.print("Disconnected from central: ");
    Serial.println(central.address());
  }
}

void updateBatteryLevel() {
  int battery = analogRead(A0);
  int batteryLevel = map(battery, 0, 1023, 0, 100);

  if (batteryLevel != oldBatteryLevel) {      // if the battery
level has changed
    Serial.print("Battery Level % is now: "); // print it
    Serial.println(batteryLevel);
    batteryLevelChar.writeValue(batteryLevel);  // and update
the battery level characteristic
    oldBatteryLevel = batteryLevel;           // save the level
for next comparison
  }
}
```