To: Robert Severinghaus
From: Team 4: Electric Juice (Ryan Sasaki, Zachary Hoover, Zachary Nakama)
Date: 4/1/2022
Subject: Testing Results Report

Project Overview

This project is called the Bioelectric Generator. The scope of this project is 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. The electric generating/charging system that we will be using is photodiodes powered by NIR light and will be placed right beneath the skin near the chest. The brain of the system that manages the energy produced by the infrared diodes will be in a 3D printed case which would be placed under the chest. This Bioelectric Generator that our team created will play an important role in the future of pacemakers as this shows the feasibility of a rechargeable pacemaker. This will stop the need for repeated and costly pacemaker replacement surgeries. In the future for this to be fully implemented in a human body the casing will need to be reduced in size and a change to biocompatible materials to be able to fit in the body.

Executive Summary

The testing phase of the bioelectric generator consisted of 9 different tests. These tests consisted of 2 matrix tests, 6 step by step tests, and a final integration test. In total approximately 60 hours were spent during the testing in order to complete these tests. The final results of testing were that the primary requirements of charging the battery and self-regulating the charging were complete, some requirements failed to pass their tests. These requirements were the temperature and the stretch goal of wireless communication.

Team 4: Electric Juice

Testing Results Report

Date: 4/1/2000

Team Members:

Zachary Hoover zdh44@nau.edu

Zachary Nakama zkn2@nau.edu

Ryan Sasaki rss272@nau.edu

Introduction

Our team, Electric Juice, was tasked with designing, building, and testing a fully implantable electric generating/charging system devised to recharge pacemaker type implants by our client W. L. Gore & Associates (future references to our client will only say "Gore"). Gore was founded over 60 years ago by Bill and Vieve Gore, where they sold Multi-tet cables. Now, Gore is an award-winning enterprise that leads the industry using their own fluoropolymer known as polytetrafluoroethylene (PTFE).

The design for our charging system is based on near-infrared (NIR) light. NIR light is used to power a series of photodiodes that are connected to an Arduino and the L6924D integrated circuit (IC), our charge management IC. Then, the charge management IC connects to the battery while the Arduino reads the voltage of the battery and transmits the voltage, via Bluetooth, to an external Raspberry Pi that displays the data on a monitor.

System Architecture

Here we have our system architecture marked with various colored squares to indicate what part of our system we will be testing. The squares encapsulate which parts are being tested with any given test. There are numbers next to each test to indicate the order they are in within the testing workbook.

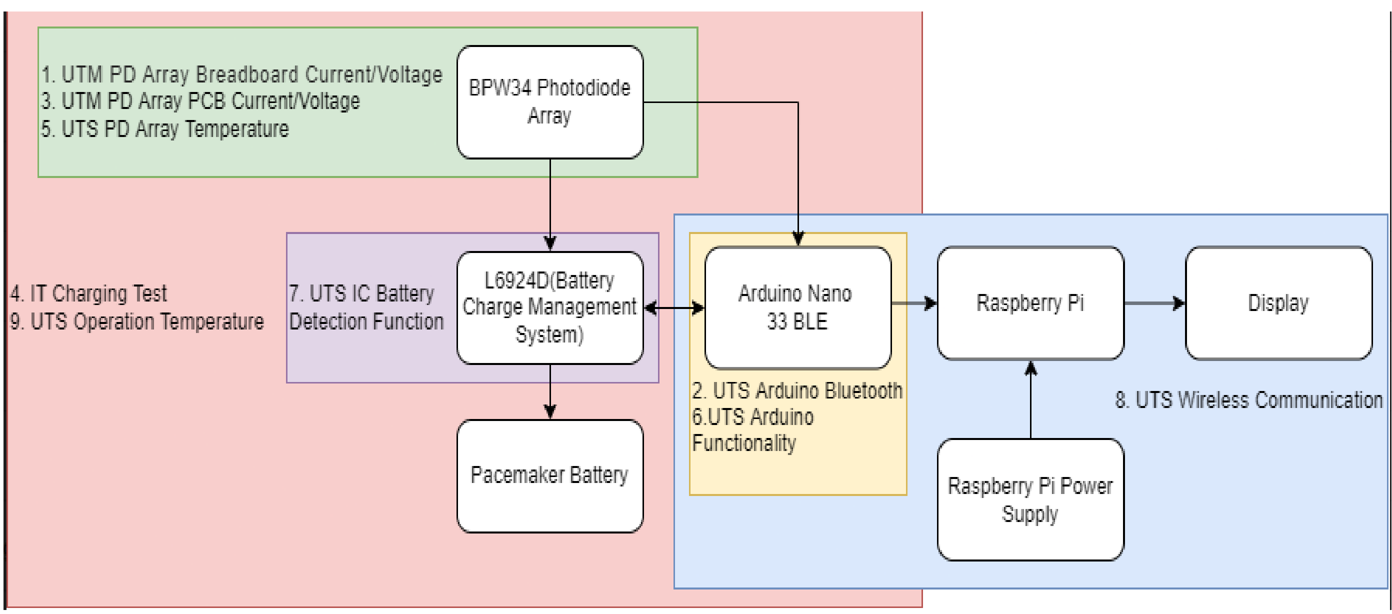

Figure 1: System Architecture with respective tests

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 2: Requirements taken from Testing Workbook

Inspect = Inspection
UTM = Unit Test Matrix
UTS = Unit Test Step-by-Step
Integrate = Integration test

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 charge 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 Lithium Polymer 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.

Types of Tests

There are four different types of tests that our team used to verify that our prototype meets the desired requirements. These 4 test types are inspection, unit test matrix (UTM), unit test step-by-step (UTS), and the integration test. The inspection test was used for 6 of our requirements which were 1.1-Size under 8cm x 8cm x 1.5cm, 1.1.1-Preferred under 6cm x 4.5cm x 1cm, 2.1-Cannot use inductive charging (no electromagnetic or coupling) , 3.2-All electrical components are completely contained within a "biocompatible" shell, 3.3-The shell of the implant does not interfere with charging method, and 5.1-Display current battery status. The inspection test was used for these requirements since these could be verified visually and did not need other methods to verify it met the requirements. The unit test matrix is used when the inputs are structurally the same and differ only in value. UTM was used for 2 of our requirements which were 4.1-Photodiode array outputs between 5-19mA and 4.2-Photodiode array outputs between 4-15V. Our team decided to use this test for these requirements since there was one factor that varied which changed the photodiodes voltage and current output. This factor is the distance between the NIR light source and the photodiodes. Here we performed UTM, keeping the infrared light source and photodiodes the same while only changing the distance the infrared light source is from the photodiodes. The next test is the unit test step-by-step which is completed by creating instructions for generating a test and checking the results which is used for flowcharts and multiple selections. The UTS was used for 7 of our requirements which were 2.3-The implant operates at a temperature below 95℉, 2.3.1-Stretch: below 80℉, 2.4-CMS does not operate if a battery is not connected, 5.1.1-Wireless transmission of battery charge level as a percentage to external display, 5.1.2-Wireless transmission of battery charge rate to external display, 6.3-Arduino can transmit data via Bluetooth, and 6.4-Arduino can turn CMS on and off. Our team decided to use UTS for these requirements since these were things that required multiple steps to create the test and verify that it met the requirements. The integration test was the last test our team used which checks that the major modules of the overall system operate correctly together. The requirements that the integration test was used for were 2.2-Does not damage battery by overcharging and 6.1-Disables IC when battery is at 3.5V. These two requirements were two major things that needed to work well with all the other components. The 2.2 and 6.1 requirements both go hand in hand with each other since the IC needs to work for 2.2 to be satisfied which is why we decided to go with an integration test for these.

Major Tests

There are 3 major tests we performed that correspond to our major requirements. These tests are the UTM Photodiode Array Printed Circuit Board (PCB) Current test, the UTM Photodiode Array PCB Voltage test, and the UTS Arduino Functionality test. 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 3*, is the plot that our team generated from the data points that we collected. We can see that the expected 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.

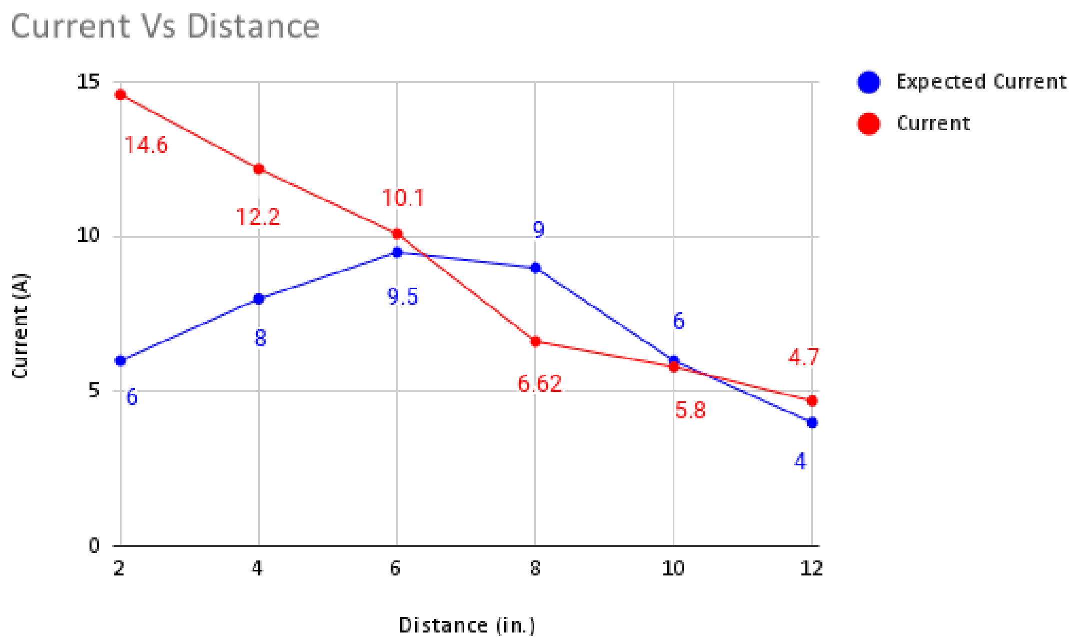

Figure 3: Photodiode Array PCB Current vs. Distance

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 the NIR lamp, 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 predict 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 4*, this can be seen in the line graph comparing voltage versus distance.

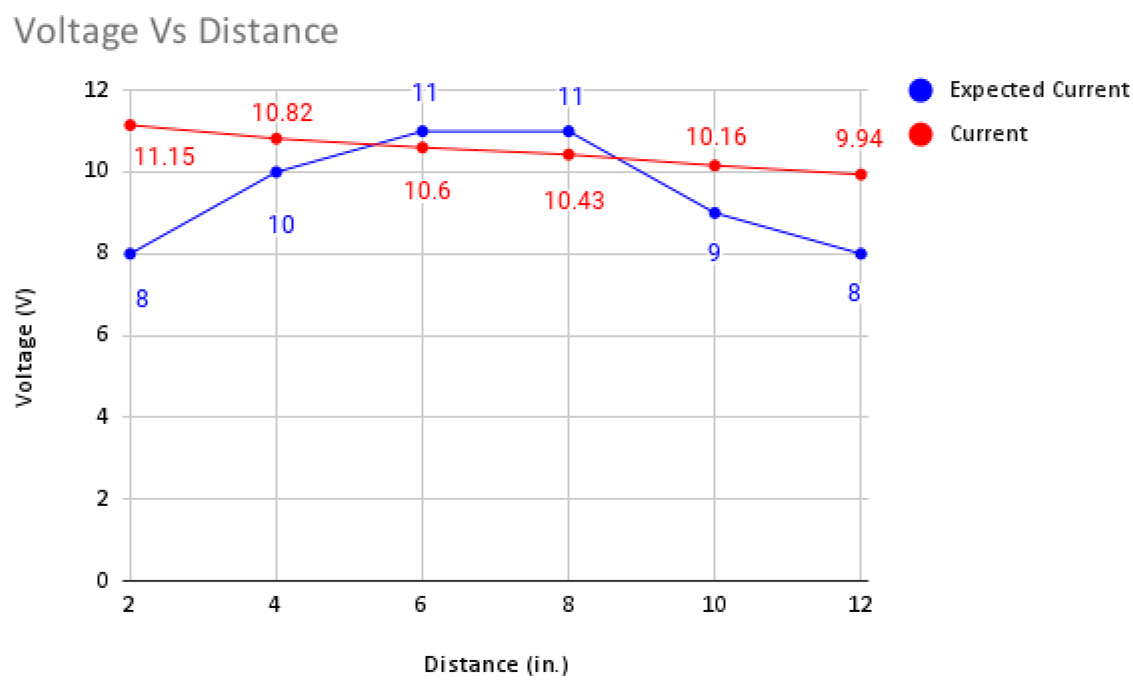

Figure 4: Photodiode Array PCB Voltage vs. Distance

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 could eventually put stress on the battery and cause the battery life to be depleted. To set up the test, we first needed 4 things to run the test which were a micro-USB to USB cable, laptop, DC power supply, and 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 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 5* below, we can see that when the SD/A5 pin is at a low voltage, the IC is enabled and at a voltage of 2V and above, the device turns off. This gave us the results that we predicted we would receive from the test. In Appendix A, a portion of the Arduino code that is relevant to this process is shown.

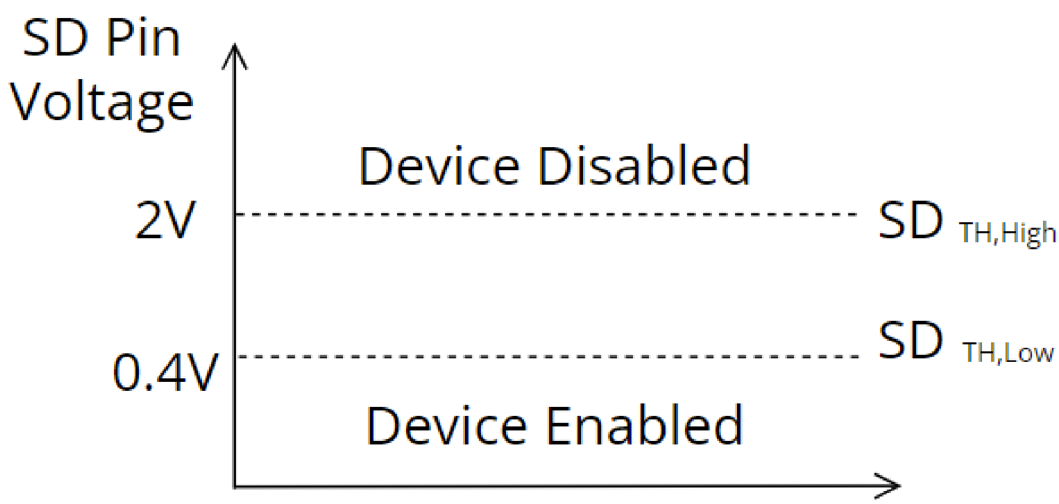

Figure 5: SD pin shutdown

Analysis of Results

When provided with a completely depleted battery, the 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.

Lessons Learned

Several aspects of the design went wrong during the development of the final product. Specifically the amount of power able to be generated using photodiodes in the surface area available is not sufficient for our design to operate efficiently. Also, the external display was unable to receive the bluetooth signal transmitted by the internal arduino. This caused a failure in the requirements for displaying information to the user. Multiple subsystems were test-fix-tested during testing. The number of photodiodes used in the design was something that was changed and optimized several times during design and testing in order to try to maximize the voltage and current while staying within the size requirements. Another subsystem that was repeatedly changed during testing was the Arduino, as the code needed to be adjusted and changed in order to meet multiple different requirements throughout design and testing.

Changes were made to the requirements during testing, as new requirements were added and others were slightly altered. Requirements were added as we determined them to be aspects of the design that we wished to verify and test in order to meet our expectations of the final product. This included adding requirements related specifically to the functionality of the Arduino rather than wireless communication as a whole, as well as altering the requirements related to the voltage produced by the photodiode array to meet a more accurate representation of the needs of the device compared to our pre testing predictions. Overall there were no difficulties performing the tests required to verify functionality of the device as the requirements were written in such a way as to be testable by design.

Appendix A

Portion of Arduino Nano 33 BLE Code

The code below shows only the portion relevant to the Arduino's task of operating the charge management IC.

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


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

}

void loop() {

  //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

}
```