# PEPA:

Programmable Electronic prosthetic Appendage

Done By:

Pascal Freij, Eric Baysinger, Elton Trimon and Tim McCarthy

### Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

## *Abstract* — The continuity in advancement of 3-D printing techniques and technology has created a new and vast section in the mechanical industry that makes it possible for the average person to be able to utilize these technologies and create any product physically at the fraction of the cost as compared to ordering and buying it from suitable vendor. Our project will employ a 3-D printer that will be used to print all physical aspects of the arm itself; the arm will be printed from the elbow all the way to the finger tips with a special design that will allow the incorporation of electric and electronic devices and circuits. With accordance to that, this project will apply most of the knowledge gained in several courses to create a programmable, technological, and sustainable prosthetic appendage.

### *Index Terms* — 3-D printing, ATMEGA2560, INMOOV, MG996 Servos, MYOWARE muscle sensor, SMARTGPU 2 LED screen

### I. Introduction

The project’s goal is to implement several electronic components and devices to produce a fully functional prosthetic appendage. In this paper, various electronic devices and elements will be integrated to produce the complete and final design.

One of our main aims is to deliver a completely functioning prosthetic appendage that can be used by any mature person to perform basic arm functions such as hand gestures and movement, as well as grabbing a multitude range of objects of different sizes. This project’s objective is to implement various sensors while being monitored by a microcontroller simultaneously to produce a bionic arm that allows any user to be able to use it without any difficulty. The last objective is to incorporate several electronic devices, parts, and elements to produce a product that can be economically attainable by middle to lower class citizens.

The specifications have been set out by the authors were the arm itself will be 3-D printed to provide a custom made physical look as well as an affordable method of production of the arm, which will act as a host to several sensors, including a microcontroller and an LED screen that will be implemented in the design. A microcontroller will be first implemented into the arm itself to act as the ‘brain’ of this appendage.

The microcontroller will analyze user data, and then perform the necessary tasks needed to control most of the functions. The flexion and extension of all the five fingers will be performed using five motors that will rotate clockwise or counter-clockwise according to which motion is required. Other sensors, such as a heat sensor, may be integrated as well to add to the technological aspects of the arm itself.

An LED/LCD screen will be added to provide an easy-to-use user interface for any individual. Some parts, such as the servo motors, will need more power than the microcontroller can provide, and as such, proper circuitry has been designed and added to provide sufficient power to all the elements that require it. The integration of all these features will result in a fully functioning arm that will perform basic hand and arm motor actions in addition to gestures.

After some research, it became clear that designing a prosthetic appendage would have been a time consuming task and as it is mainly a mechanical aspect of the project, we opted out of designing it and instead, looked for other sources to obtain STL files of an arm that was either an open source or the owners provided permission for using it.

After obtaining the files, the arm itself needs to be printed, and buying a 3-D printer or using private owned printers has been proved to be costly as purchasing a printer requires a good monetary investment, while private companies that 3-D print object for consumers charge each project by the 3 dimensions (x, y, and z) and the total weight of the object as well as the total amount of time it took for the project to be completed. The choice of servo motors was decided as it would perform the necessary task, and the affordability of the motor itself played a part into choosing it. Research into durable lines was concluded by having to either decide between using fishing lines or guitar strings as a mean to connect the servo motors to the fingers individually.

### II. Electronics And Physical Parts Chosen

The LED screen chosen was a Vizic Technologies 320x240 screen called SMART GPU 2 as it fit perfectly for our application and price range. It has a variety of capabilities to choose from for implementing into our design, such as supporting games, programming a calculator, or even a clock. It also includes a slot for a mini SD card in order to add music, pictures, or even MP4 videos depending on the nature of the design. The purpose of this screen in our design is to realize a UI that is simple in architecture and easy to use for any consumer.

The Arm itself is designed by INMOOV. An open-source life sized 3-D printed robot, and we are just using their arm designs. We sent the STL files to ASME (American Society of Mechanical Engineers) at UCF, and had them print the arm at the beginning of last semester.

The only other 3-D printed part that we designed ourselves was the housing component for all the electronics. We have fully designed it from scratch using Solid Works according to the measurements we have taken of the arm from INMOOV with relation to its size. It mounts on perfectly on top the arm with sufficient space to house all the electronics we need and more.

The servo motors chosen where the MG996 as they will provide us with the precision and control that we want to achieve in order to perform all the gestures we want as well as other hand motions such as grabbing and releasing. The wires used to connect the servos to the arm and back to the servo are beading strings, as our research done last semester proved that they are the most efficient price wise as well as durability wise.

The microprocessor chosen was the ATMEGA 2560 for several reasons. Mainly compatibility with the LED screen that we have chosen as well as the RAM capacity compared to other models. It can control all the necessary components as intended. It is also programming compatible with the MYOWARE muscle sensor that we plan on implementing.

Choosing the correct battery played a critical role in providing realistic operation to the arm. Figure 1 showcases the battery decision and how the group decided on the final battery design. When our group looked at batteries, we had to make sure that the battery had the specifications for the arm. Our group based our decision on toxicity, capacity, and the overall cost of the battery. A lot of batteries are toxic and have to be properly disposed. For instance, Alkaline batteries contain potassium hydroxide or sodium hydroxide. If the chemicals came out of the battery, the compounds can cause liquefaction necrosis of the skin. These batteries are immediately ruled out since they can cause harm to the user of the Bionic Arm. If these batteries leaked in the arm it could cause damage to important components in the arm. Since the bionic arm is connected to an area where a person had an amputation, the toxicity level could affect the user more if the battery was pierced for any reason. Furthermore, the capacity of the battery plays a major role when a battery was picked. Figure 4.2.1.a shows the capacity of all the 9 volt batteries that were researched for the arm. By looking at the graph, our group could see that two batteries had very low capacity, and two had very high capacity. Even though the Lithium Ion had the highest capacity, our group decided that it would not last long enough for normal use or demonstration purposes. These 9 volt batteries were smaller than the RC batteries.

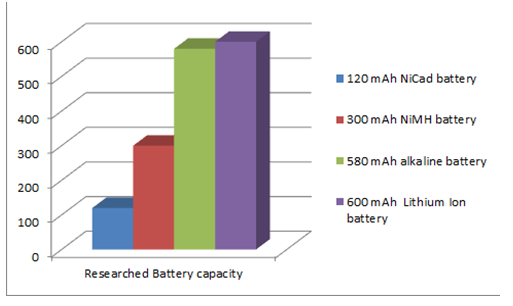


Figure 1. Battery Comparisons for Power Circuits

### III. Project Details

The whole design can be separated into two layers, the programming and physical/electronic layers. The programming layer required sophisticated programming knowledge in order to combine several electronic elements and devices to work coherently and effectively to produce a desired output. The second layer, the physical layer, consists of all the electric and electronic elements and devices that we have used to realize this design. It is worth mentioning that the programming aspect of this design was foreseen as the most difficult part, regardless if it seems to come as second nature after months of practicing. Figure 1 showcases the primary components of the project and how they are combined together to create a cohesive arm.

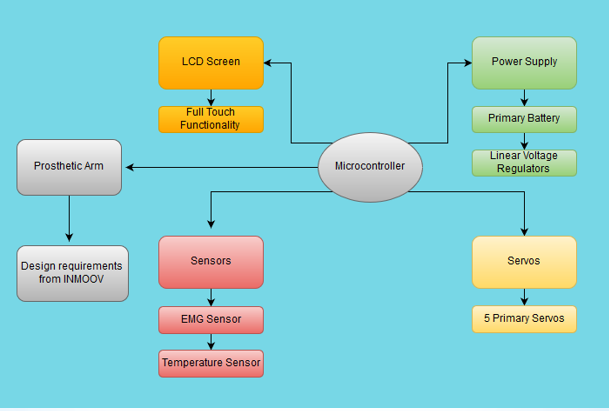


Figure 2. Overall Design Layout of Project

*A. Programming Layer*

The process of programming for the project combined several different components and techniques in order to both fulfill all of the requirements for the finished project and to ensure that the program runs efficiently and with little to no computational errors. The first of the components involved with the programming was the choice of languages used. Our choice for the programming languages was decided based on the compiler chosen, and the microcontroller used. Namely the Arduino IDE and the ATMEGA328P. The languages we used are, the Arduino variant of C++,C++, and C. C++ and C were used within the libraries and header files used to communicate with the drivers for our electronic components. The Arduino variant of C++ was used to write the main program itself. The Arduino IDE was used because of the use of the Arduino UNO as our development board and our use of the UNO in order to flash the microcontroller used on our final PCB.

Other major part of the programming for the project was the design of the main program itself. In designing the layout of the main program there were several major factors that had to be considered. The first and most important of these was the integration of our peripherals; namely the SmartGPU2 touchscreen and the MYOWARE muscle sensor. The touchscreen, with its own built in controller, has a large set of commands needed to program it, stores in a large C library, and requires constant polling from and a specific set of baud rates from the microcontroller in order to function correctly. The MYAWARE sensor also comes with a specific library in order to read correctly. The impact of the muscle sensor on the overall programming was rather small as everything need to read from it was very low impact. The touchscreen on the other hand has a major impact on how we designed the program. Because of the touchscreens requirements we designed the program as a continuous polling one consisting of a single loop that runs the entire time the program is running. Also in order to do this we designed out all interrupts. The loop checks for any touches on the touchscreen and then updates the state of the screen and the program as needed. The muscle sensor is checked right after the touchscreen each time through the loop.

Another major design choice with the program design was based on random access memory usage. This was caused because of the servo motor libraries that were used in the programming. When a servo object is created inside of the program it uses up some of the RAM available which is ok. But when the servo object is connected to a specific pin on the microcontroller there is another object created in RAM. This would normally also be ok but in our case if we have all 5 servo motors attached and are running the touchscreen we will run out of space in RAM. This would cause the program to not function as intended so we had to design a work around into the program to prevent this from happening. Due to the nature of the touchscreen being always active the work around would have to be in how we handled the servo motors. In the end we decided to disconnect the servo motors from specific pins when they were not in use, and when needed we only attach 2 at a time. This design choice completely removes all of the RAM issues we would have had at the expense of a slight slowdown in operating speed of the arm. This slowdown was considered an acceptable loss when compared to the program not working at all.

Other choices we had to make with the design of the program were based on the needs of our graphical user interface for the touchscreen and the amount of user options that we wanted to be available to the user. As we choose for there to be six hand movement options for the user we had to implement six buttons into our program design. This choice is what uses up most of the RAM available to us as each button is a structure that contains its location, size, color, text, function, and other properties related to reading from them. In addition to RAM usage this design choice also increased startup time significantly by around 2 seconds. This delay is caused by communications with the touchscreen and having to use delays when sending commands to it in order to have them properly understood. We stuck to this design choice despite the delay it creates as we believe that more user options is worth an exchange of 2 seconds every time the program starts up. This is due to the relatively rare situation of starting up the program.

The last major choice made within the programming layer was the implementation of power saving options, or in our case the lack there of. One of the main concerns we had with the programming is the usage of as little power as possible. This is done to increase the running time of the arm to as high as possible. As we designed the program we tried multiple power usage reduction techniques, such as reducing the baud rate, reducing the amount of I/O checking done, and using low power modes. But due to the other requirements of the program, and the fact that most of the power is used to move the servos which we cannot control, we ended up deciding that none of the power saving options available for the program would be worth implementing. Other factors that went into this decision was the shortage on available RAM and possible increases in delay between commands creating a worse user experience.

*B. Physical and Electronic Layer*

While on paper this section seems straight forward, it is one of the most important aspects in this design. This section discusses the electronic elements whose purpose was to create a duplex communication between the MCU and every device. This layer consisted of every physical electric and electronic elements and devices that we will be embedding into the final design of our project. We will begin with discussing the prosthetic arm design itself.

INMOOV is an open source, 3-D printed life size robot, were we asked for permission and received their arm design. We printed the parts using ASME’s printer, and obtained the disassembled arm last semester. This semester we assembled the arm, and made sure it became more sustainable as the 3-D printer ASME used had created multiple imperfections in the design. After fixing all the parts, we finally realized biggest physical constraint in this design, the arm itself. The arm was designed in a way to hold 5 servo motors, with room to allow wires to go through the arm, and back in order for a motor to control basic arm movements. In addition, it was also designed to be split in five parts, the fingers, palm, base of the palm, and lastly the arm split in two parts. The main reason was to simplify the assembly process and the process of connecting the wires from the motors to the fingers and back. Figure 2 showcases the look of the arm upon the conclusion of the arm being printed.

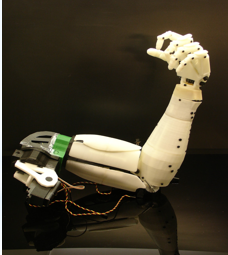


Figure 3. Finished Arm Construction Overview

The 3-D design that we created was the MCU housing unit that will be placed on top of the arm. It will hold the MCU, as well as the LED screen, sort of like an oversized watch. The design was perfectly fitted and adjusted as we needed in order to have a fully incorporated physical prosthetic that did not look out of order. The design also had perfectly designed whole in order for wires to pass through our design and into the arm in order to power up the system as a whole. As mentioned earlier, the Arm was designed in a way to allow for wires and motors to fit in and both will be discussed below.

We have chosen to use the MG996 Servo motors for two main reasons: Control and precision. There are many methods in which we can control our servos. Because we are required to use a microcontroller, we limit our control options to microcontrollers. There are many much simpler ways to control servos by using more “mini-computers”, but our group is more interested in how to use embedded systems for control.

When controlling servos with embedded systems, we use C programming to send output signals using the Timer. There are multiple servo control points that must be addressed. First, the power supply must be filtered so that we limit the usage of noise. Second, when connecting the signal wire to the microcontroller the signal must be filtered by using a simple resistor. This allows there to be a slight buffer to protect the microcontroller. Next, servos must be powered from a separate power supply. There are multiple reasons this should be done, but the biggest is that the servos are powered off a higher voltage. We don’t want the servos to only be fed 3V. The more voltage that is supplied to the servo, the greater torque that is received (up to a certain point) and finally, the servo and the microcontroller must share a similar ground.

One of the key reasons as to why we must use a servo versus a typical DC or AC motor is that with a servo, we have precise control over the position of the servo. Servos are precise due to the way they function. . First, we have a minimum pulse of about 1ms which keeps the servo at 0 degrees. If we increase the pulse to around 1.5ms the servo goes to neutral position and changes to 90 degrees. Finally, if the pulse is around 2ms, the position of the servo advances to 180 degrees. These pulse widths can vary and the position will change depending on the “home” position of the servo. Obviously depending on the time that’s sent we can achieve any angle that is desired. We will program with these ideas in mind to get exact hand gestures each and every time a “program” is selected. The user would not want any ambiguity in which gesture they are calling for so it is critical that the servos that are selected as well as the microcontroller can provide the precision that we are looking for. The above meet these requirements. Another important point is that the servos must always know how to go back to the home position. Once the arm picks up the item and eventually releases it, the servos need to return to their home position and allow the object to be dropped off. There are 5 servos included in the arm, so there will be 5 servos that all need precision given directions by the microcontroller.

After researching, we realized that having bead strings connected from the servos to the finger tips and back would be the most sustainable method. By sustainable, we mean that the bead strings we chose have the least occurrences of tearing up or breaking down due to being pushed and pulled over a long period of time, unlike guitar strings or fishing lines that have a lower life span with regards to the stress endured due to the servo motors pushing and pulling. The next part to discuss is the EMG sensor.

The bionic arm needs to communicate with a person’s arm. In order to realize that link, electromyography (EMG) sensors were installed where a person’s arm meets the bionic arm. EMG can also be considered muscle recording sensor. These sensors are usually used to for medical research and the diagnoses of muscular disorders. The sensor is able to detect signals when a person’s muscle is moved. By detecting this signal the bionic arm can communicate with the arm. The signal will be received by the microprocessor and will perform a programmed action depending on the motion.

By programming the servo motors, the arm can be programmed to recognize specific and unique moves that an individual might want to perform with the arm. The EMG sensors can sense signal frequencies between 0 and 500 Hz. The interesting thing about this signal is that it comes directly from a person’s brain. In order for the body to perform an action, a signal has to be sent to the area of the body where the action is desired. These signals are sent through the nervous system in the body. The nervous system can be seen as signal going through a wire in a circuit. The EMG receives this signal and the microprocessor will determine which action needs to be taken.

To describe this process with regards to our project design, we will have two inputs from the user. The user first chooses a category on the LED screen, allowing the MCU to know which action to take, but to wait for a secondary input from the user, the muscle sensor. So once, a user chooses an option, thumbs up for example, the MCU would wait for the user to stress on his/ her bicep in order to allow the MYOWARE muscle sensor to transmit to the MCU. Once the MCU receives the analog signal and decodes it to digital, the result would be having the thumbs up motion. The last feature to discuss in the physical layer would be power and will be discussed below.

The Arm cannot be powered directly since it is a portable unit. Since it needs to be used in areas where there might not be power, the unit has to be wireless and be able to last a decent amount of time. Batteries will be utilized in order to power the bionic arm. The power usage of the arm is expected to be really high since the unit has 5 servos, an LED screen, and a microprocessor. The batteries used for the arm have to be rechargeable batteries in order to save money when using the arm. The batteries considered are 9 volt batteries or RC batteries. The kind of battery that will power the bionic arm is NiMH. Figure 4 displays the power system developed for the arm.

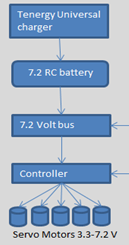


Figure 4. Power Diagram to Servos

Voltage regulators will be used to regulate the voltage to the desired voltage for the microcontroller, LED, and the servo motors. The LED battery life should last longer since it uses less power than the 5 servos when active. In order to save power when the LED screen is not in use, our group will program it to go into low power mode when it is not active. The microcontroller in the bionic arm will also be programmed to know when the sensors are not needed in order to save power. The main goal is to save power so that the use of the arm is optimal; therefore, most components will be in sleep mode.

The LED will display the battery life for the servo motors and the screen itself. Furthermore, the user should take care when using the bionic arm to carry heavy objects that might overload the servo motors. If the bionic arm is overloaded, the servos will not run smoothly and use a lot of power so that it can complete a normal programmed action. The user should use the bionic arm for its designed purpose and pay attention to its weight limits. We expect the arm to operate at optimal consuming power normally if it lifts objects between 1 to 1.5 pounds. If the bionic arm is used to lift normal light items and use programmed hand gestures, the battery life is expected to last 2 to 4 hours with the high capacity battery that we have purchased.

The programming of the microcontroller is the most important part in order to save power. The main goal of saving power is to maximize the use of bionic arm after every single charge. Power will also be saved when the bionic arm is programmed to know when to lock and hold an item for an extended period of time. For instance, the user could be holding a can, fork, and a pencil. The user will be able to let the microprocessor know via the LED touch screen that he or she wishes to keep a state that the arm is in after an action is performed. This will indeed make the bionic arm smarter when saving power.

The EMG MYOWARE sensor will need Voltage regulators so that it’s powered with a 9 volt battery. In order to lower the voltage, 5 volt regulators will be used to reduce the voltage. For the EMG circuit, 5 volts and negative 5 volts is required to make the servo work. Our group will not need to purchase regulators for the servo motors. The RC battery is 7.2 volts which is in the range of 3 V to 7.2 V.

The LCD screen will also need to be powered. The LCD screen that will be used is the GPU 2. In order to power the GPU 2 LCD screen, a 3.3 voltage regulator will be needed to lower the voltage. The regulators will be placed in between the battery and the components. The responsibility of the regulator is to regulate the voltage for every single component in the device. The LED touch screen will need a regular for 3.3 volts. Since the touch screen is so sensitive, it is very important that the correct voltage be applied to it in order to avoid damaging such an expensive part of the Bionic Arm. In addition, the servo motors do not require a regulator since the servos can operate at between 3.3 and 7.2 Volts. The battery for the servos is 7.2 volts in this case.

Now that we have discussed both layers that will be implemented, it is now time to discuss our own circuit. In the next section, we will discuss our very own MCU design that will be used to realize the goals that we have set at the beginning of this project.

### IV. Eagle Design

In order to connect all of the different components of our project, the Eagle schematic based software is used to connect everything together. Eagle, first allows the user to build the circuit diagrams in a schematic based format and then further allows the construction of a PCB based format. The design for this project is simple in that it only requires two voltage regulation circuits, a microcontroller, and a multitude of headers to provide a plug-in-play application for servos, motors and any other extra peripheral sensors or motors required for the project. Due to the DIP layout of the microcontroller chosen, there is no reason to add additional circuitry to program the device. Rather, the device will be programmed using a development board and plugged into the final PCB design. This allows for a significantly simpler layout for the prototype being created.

Because the arm will be using 5 servos potentially all moving at the same time, power efficiency is a concern for the project. To avoid exhausting the battery, switching voltage regulators were chosen over linear voltage regulators. Two designs were needed here: 3.3V regulator and 5V regulator. Their primary designs can be viewed in Figures 3 and 4. One of the most important aspects when designing the regulators was finding appropriate components to not only handle the appropriate current regulations delegated by the servos, but to find specific components that will be solderable onto the board using a reasonable soldering iron. Once the components were determined, the final schematics were put together as seen below.

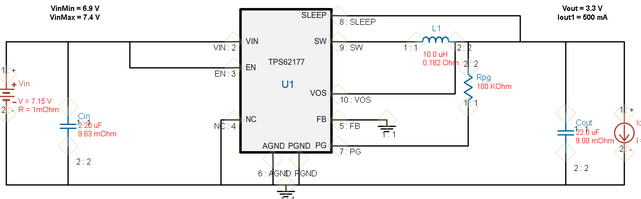


Figure 5. 3.3V Regulator

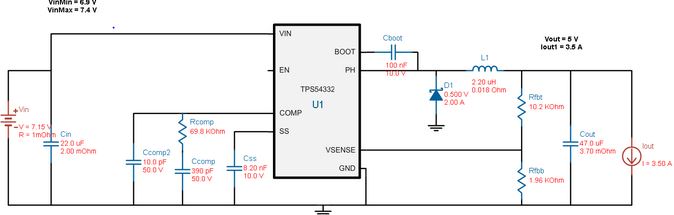


Figure 6. 5V Regulator

In order to actually build the circuit in Eagle, each component was selected and placed around the microcontroller in their respective locations, the NET tool was used to combine and create the final circuit. Two basic “bus” configurations were designed to provide the 5V to the servos and about 3V to the microcontroller and LCD touchscreen. To step down the 7V provided by the battery, a 3V voltage regulator is built into the circuit to provide corresponding power.

The control lines on the servos are connected to these pins so we have full control of not only the servos motion, but ultimately the hand’s motion. If needed, the EMG sensor’s control line is connected and can fully control and allow for analog to digital communication through this pin.

Finally, the LCD screen allows for use of either just TX/RX pins or TX/RX pins, ground and voltage. We will use the 4 pin configuration so that we can house all of the critical connections with one header. The NET tool is used to connect the corresponding pins from the MSP432P401 (pins 9.6 and 9.7). The other pins connect to the voltage regulated 3V bus.

Coursework has always used a Launchpad or some other development board to program a microcontroller. The real world requires the use of programming a standalone chip. Programming a standalone chip requires the use of JTAG or external device.

In order to simplify the process, JTAG is avoided by programming the chip on a breadboard, ensure its

Our completed schematic can be seen below with all of the features mentioned above. Note that this schematic is a draft and can be updated with the project. Once the components are placed on a breadboard and proven to work, the PCB can be constructed and sent to OSH Park to be built.

VII. Conclusion

The project really came together during our design phase. From the 5 servo design choice to the LCD touchscreen, our group believes that our project stands out from other groups designing prosthetic limbs, particularly groups working with Limbitless.

There were further aspects of the project outside of our typical studies that were learned as well. Working in a group is a critical part of any engineering project so it’s important to have the opportunity to do so while in college. We were able to delegate specific tasks to each team member and apply different portions of the project to the member’s skill sets. Deadlines were set and each member was expected to meet the dates that were provided. If a team member wasn’t able to complete their portion of the project, other team members were expected to pick up the slack and complete the work regardless of the situation or why the team member wasn’t able to complete their portion of the project. When 4 people write a paper, it can introduce challenges with ensuring redundancies and similar information. Special care was taken in assigning each portion of the paper so the paper flows as seamless as possible without the reader realizing there are multiple authors.

When writing the paper, our group used pictures and tables from outside sources. It was required that we ask for permission to use their work and insert their approval within this paper. These kinds of ideas apply throughout the corporate world and it was a valuable life lesson in ensuring that everyone’s work is properly cited and formatted. It also provides learning skills in sending professional emails to, on occasion, large employers or corporations that we could someday work for.

All of the components of the project have really come together during our design phase. From the 5 servo design choice to the LCD touchscreen, our group believes that our project will really stand out from other groups designing prosthetic limbs, particularly groups working with Limbitless. Senior Design II will demonstrate everything we’ve worked for this semester and our hope is that Senior Design II will really be able to provide a true capstone for the entirety of the Electrical Engineering degree here at The University of Central Florida.

### Acknowledgement

The authors wish to acknowledge the assistance and support of all faculty and instructors at UCF as well as Jonathan Stagnaro for his assistance in design of the Eagle based components and PCB direction.

### References

[1] Davies, J. H. (2008). *MSP430 Microcntroller Basics.* New York: Elsevier.

[2] Tenergy. Tenergy Universal Smart Charger for RC/

Airsoft Battery/ NiMH/NiCd Battery Packs (6V -

12V)”

[http://www.amazon.com/Tenergy-Universal-](http://www.amazon.com/Tenergy-Universal-          Charger-)

[Charger-](http://www.amazon.com/Tenergy-Universal-          Charger-)

[3] Inmoov “Hand and Forarm”

[http://inmoov.fr/hand-](http://inmoov.fr/hand-              and-forarm/)

[and-forarm/](http://inmoov.fr/hand-              and-forarm/)

[4] Vizic Technologies. “SMART GPU 2 2.4” TOUCH”

### Engineers

**Eric Baysinger:** Eric is currently working at Lockheed Martin as part of the College Work Experience Program. He will graduate with a B.S in Electrical Engineering.

**Pascal Freij:** Pascal is currently working at Jammal Engineering as a Lead Project Manager. He will graduate with a B.S. in Electrical Engineering from UCF and will continue on to achieve a Maste’rs Degree.

**Tim McCarthy:** Tim is currently studying Computer Engineering at UCF and will graduate with a B.S. in Computer Engineering.

**Elton Trimon:** Elton is currently working at JBT AeroTech as a product engineer intern. He will graduate with a B.S. in Electrical Engineering.