ACDC - A Helping Hand

Akash Jinandra, Carlos Cuesta, Devin Defond, Chang Ching Wu

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

***Abstract* — This paper presents the design methodology utilized to realize the telerobotic arm that was designed by Group A. One user wears an exoskeleton with sensors on each finger, elbow, shoulder and forearm. It transmits the data to a robotic arm that mimics the movement of the user. This paper is broken into the following sections: (1) mechanical construction, (2) embedded processing, (3) power distribution, (4) communication and (5) software controls. This paper discusses the implementation of both the hardware and software. The primary task of the system is to mimic the user as closely as possible.**

Index Terms — Atmega328P, Bluetooth Communication, tele-presence robotics, safety.

I. INTRODUCTION

There are many environments. The motivation of the project came from watching the movie surrogate. We noticed that the surrogate robot can go through many hazardous conditions and environments that humans cannot. Obviously we do not have the technical knowledge and resources to create these full scale robots with that much detail. What if instead we could control the robots with our actual movements instead of using a joystick or controller? Also we could not make a full scale robot let alone control it during our 1 semester building phase of senior design. So we thought of the minimum viable product that would be the most useful. We settled on a robotic arm. This is arguable one of the most useful parts of the human body, the dexterity of 5 fingers and the four degrees of freedom of a whole arm. We decided the control will have a form factor similar to a sleeve, which will contain sensors on each finger, wrist, and major joint of the arm. The finger sensors will be using accurate flex sensors that change resistance as the finger flexes. The sensors will also be used on the elbow as well as the palm and the wrist. These sensors will add gripping capability. Gyroscope sensor is what we plan to use on the wrist to add degrees of freedom. All of these sensors will be sewn into a fabric glove. The sensors will be wire onto a microcontroller, MCU, such as Arduino or a MSP430 board. We are still deciding whether the communication will be local or will be over the internet

The Arm itself will be mounted on a table from the elbow. It will be able to move up and down and rotate 360 degrees horizontally. We plan to use six micro-servo motors for the fingers and the pisiform, to give each fingers and the pisiform the ability to open and close with the similar dexterity as an animatronics' hand. Two standard-servo motors will be use to give the palm of the hand the ability to fold up and down and swing left to right, by pushing/ pulling moment arm attached to the wrist at different combinations. For the elbow, we plan on using a stepper motor for the beneficial factor of high torque and higher precision of micro-stepping. But, since there is no feedback for a stepper motor, a feedback sensor, hall-effect encoder, should be attached to the rotary axis of the elbow to allow accurate measurement for feedback.

The applications of this project are limitless. We can attach this arm to a robot and use it for bomb disposal. The bomb technician will not have to learn a new system and can use his instincts without hesitation. We can also use it in a situation where there is radiation and our arm has to operate things that humans usually operate, such as controls, switches. A more domesticated version of this would be mounted on a desk. This can be used for support for IT. For instance, an elderly person who can’t type for themselves, or don’t know where to click the mouse. Or maybe helping engineers with circuits, if they don’t know how to connect something properly, a seasoned engineer can help them with ease.

II. EXOSKELETON

The exoskeleton is a key part of the project that is needed to capture the motion of the user that is trying to do the task. The design for the exoskeleton was found online and was made by Alex Czech. The original design included joints for both arms with rotating points at the each finger, elbow, and shoulder. This design had to be modified to allow for this project’s specific application. For instance, this project only required only one arm. Also with the previous design, it was very difficult to capture shoulder movement due to the original joint being a ball joint. Mounting a sensor would be very difficult. To fix this, the elbow joint from the second arm replaced the shoulder ball joint. The connection from the hand and wrist was also disconnected to allow for wrist rotation.

With the new design it was possible to capture rotation from both the elbow, shoulder, fingers, wrist and forearm. The method of capturing the motion or each joint was done in two ways. For the fingers rotary potentiometers were used, as a high resolution was not needed for this, because the hand that was given to the team did not have that high of a resolution. For the elbow and shoulder joints, 4000 step quadrature encoders were used. These are the same encoders that are used in each motor that is inside the arm. To capture motion effectively, a potentiometer was mounted to each knuckle on each finger, and the encoders were mounted to both the elbow and shoulder joint. A picture of the motion capture exoskeleton is seen in Fig. 1.

Fig. 1 Exoskeleton

III. HARDWARE

In this section two different aspect of hardware in regard to this project will be discussed. These aspect will be parted into subsection A, which goes over the mechanical hardware, and subsection B, which goes over the electronic hardware.

A. Mechanical Hardware

The mechanical hardware of this system is constructed and provided to this project by HP. The mechanical arm has capability of 5 degrees of freedom, DOF, not counting the end effector. The shoulder joint have 2 degrees of freedom, rotation on the y-axis and z-axis in relation to the mounting stand. The elbow have 1 degree of freedom, rotation on the parallel axis to the shoulder. The forearm have 1 degree of freedom, rotation on the perpendicular axis to the rotation plane of elbow. The wrist have 1 degree of freedom, rotation on the perpendicular axis to the rotation plane of forearm. Although, the mechanical arm is capable of 5 degrees of freedom, only 4 degrees of freedom is utilize to realize the objective. The rotation on the z-axis of the shoulder would be disregarded.

Each degree of freedom is achieved through the usage of 24V DC motor equipped with corresponding reduction gear. The reduction gear set is different for each degrees of freedom due to the dynamic aspect of the mechanical arm, some degrees of freedom would need to output more torque than others. For example, the degree of freedom on the shoulder responsible for the rotation in the y-axis is capable and required to output more torque in order to manipulate the links and end effector.

The position of each degree of freedom is tracked through the usage of rotary encoder sensor. For each degree of freedom, there is an integrated rotary encoder mounted on the rear shaft of the DC motor. The position of each degree of freedom is interpolated through the processing of rotary encoder data by the electronic system.

The end effector of the arm is a 3D printed hand with individual movement capability for each of its fingers and thumb as shown in Fig. 2. Each finger is manipulated by a micro metal gear servo with metal braided wire. The thumb is manipulated by a standard servo and metal braided wire.

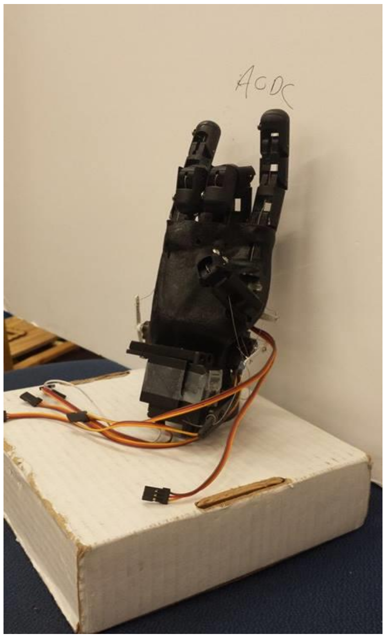


Fig. 2 Mechanical Hand

B. Electronic Hardware

The electronic hardware of this system is divided into two separate systems: the sensor sleeve side houses one of the embedded electronic system. While the mechanical arm side houses the other embedded electronic system. Both embedded electronic system is equipped with a USB-to-Serial interface which enable the capability of updating code and debugging each system. A main microcontroller unit which process the corresponding data on board the embedded system can be found on both sides. Along with several sub-level microcontroller unit dedicated to processing position data and or control, as shown in Fig 3.

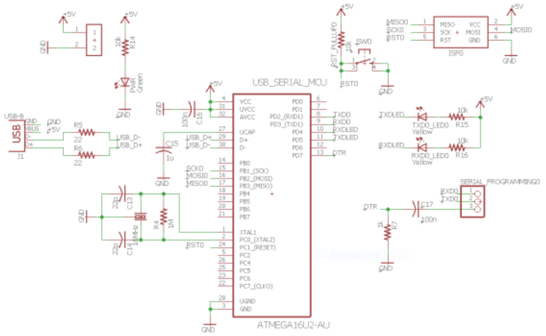


Fig. 3 USB-Serial MCU

Implemented on the Sensor Sleeve side is a communication block, a microcontroller handling the overall processing, and two additional microcontrollers interpreting the two rotary encoders. The main processing block tracks position of user's fingers, forearm, elbow, and shoulder. The two interpreting block process the rotary encoder sensor data and send the processed data to the main microcontroller. The overall schematic of the Sensor Sleeve side is shown in Fig. 4, while the printed circuit board layout is shown in Fig. 5.

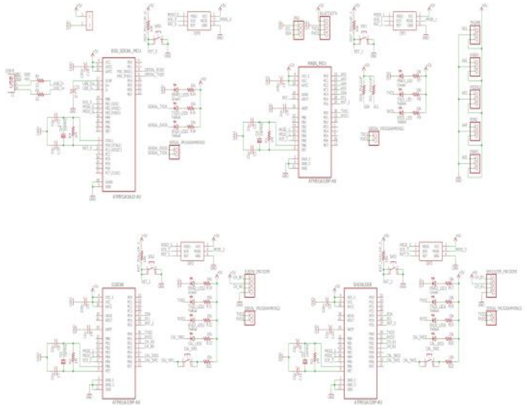


Fig. 4 Exoskeleton Embedded Hardware Schematic

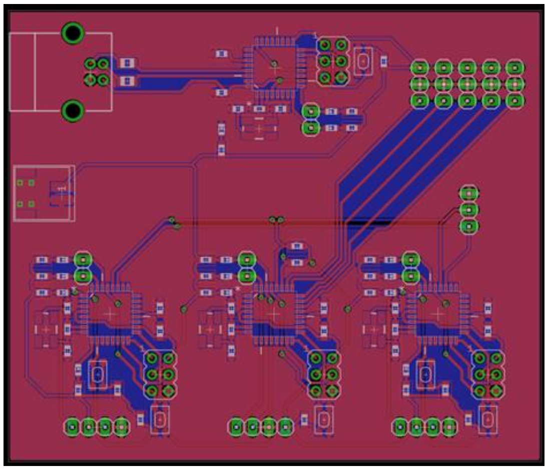


Fig. 5 Exoskeleton Embedded Hardware PCB

Implemented on the Mechanical Arm side is a communication block, a microcontroller handling the overall processing, and 4 additional microcontroller handling the 4 degrees of freedom. The main processing block handles the position data from the Sensor Sleeve side and send to the corresponding reaction network. The reaction network are shoulder processing block, elbow processing block, forearm processing block, and wrist processing block. Each processing block forms a feedback loop using the rotary encoder sensor data and motor driver. The overall schematic of the Mechanical Arm side is shown in Fig. 6, while the printed circuit board layout is shown in Fig. 7.

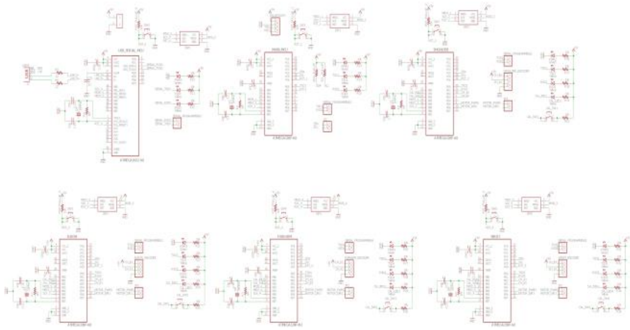


Fig. 6 Arm Processing Schematic

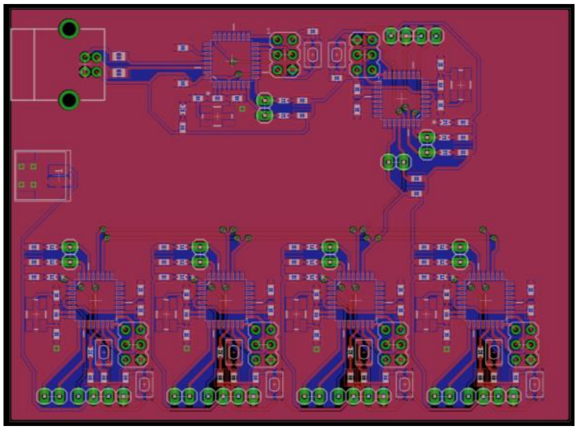


Fig. 7 Arm Embedded Hardware PCB

IV. COMMUNICATION

The nature of the project, A Helping Hand, is to allow 3D printed exoskeleton arm to wirelessly connect to a robotic arm, and then to have said robotic arm mimic the movements of the exoskeleton. Due to the need for a wireless connection, it was necessary to find a wireless communication standard that would best fit the needs of this project. After much research between many different standards (Wi-Fi, Bluetooth, Sub-Ghz, etc.) it was found that Bluetooth would be the communication standard that most aligns with the project's needs. The reason for choosing Bluetooth was due to the ease of implementation, the low cost, and the speed that Bluetooth provides.

The Bluetooth device that is being used is the HC-05 and HC-06 Bluetooth modules. The reason for the two different parts is due to the way how these Bluetooth modules work. In order for the modules to connect and then begin sending data over wirelessly, it needs to be set up in a way that there is both a Slave module and a Master module. The HC-06 is the slave and the HC-05 is the master. The master module is attached to the robotic arm, waiting to receive the data being sent from the slave module (which is attached to the exoskeleton), and then the data that was received will be processed by the microcontroller and then delivered to the required motors in order to move the arm.

Now for the code that operates the communication system for A Helping Hand. From what was stated previously, the master module is on the robotic arm and the slave module is attached to the exoskeleton, because of this the slave module will be the one that will be receiving all the measured data from the user's inputs and then transmitting that to the master module so that data can be processed and then used for the robotic arm to mimic. In order to do this, an integer array was constructed to host all of the user's inputted movement; however, this did not work during the initial test runs. The reason for it not working was found to be caused by the way that the Bluetooth modules were transmitting the data. The HC-05 and the HC-06 transmits data as a single byte, a byte is composed of 8 bits, and with only 8 bits to work with the largest number that was being transmitted was 255. This was unacceptable due to how it will impact accuracy (the potentiometers can go up to 1024 resolutions, thus there would be a 75% drop in resolution, the encoders can go up to 4000 resolutions which would be a massive 93.75% drop in total accuracy).

The solution to this problem was to find a way to transmit the recorded integer values (an integer is 2 bytes) in a form of two separate bytes. The first step was to create a byte array that would have a buffer size that is twice the size of all recorded integer values. After the creation of this byte array, what was then needed was to find a way to break down the integer values into 2 bytes each. Thankfully, Arduino has a command that can capture the first 8 bits (which are consider the high bits/byte) and the lower 8 bits (which are consider the low bits/byte). After capturing both the high and low bytes, both bytes would be recorded in numerical order within the byte array. This byte array would then be transmitted from the slave module to the master module for processing.

Once the byte array was transferred from the slave module to the master module, it was time to decrypt the byte array into an actual integer value. The equation used in order to recreate the original integer value is (1)

$$\text{int value} = \text{highByte}*256+\text{lowByte}. \quad (1)$$

The reason behind multiplying the high byte by 256 is to tell the microprocessor that the first byte is to be shifted to the left by 8 bits which would then allow the high byte to actually represent its intended value. After adding both the high and low bytes together, the sum will be equivalent to the original recorded value. Once all the integer values have been recalculated, the master will then take these calculated values and send it to all of the servos, which will then cause the robotic arm to move. This process has a visual representation found in Fig 8.

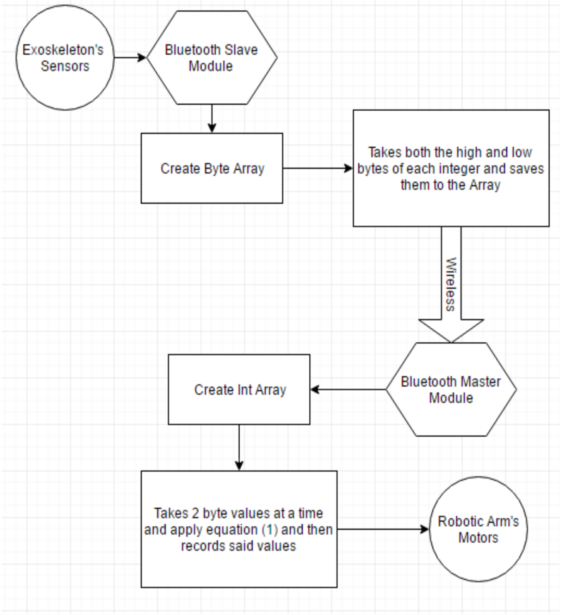


Fig. 8 Communication Block Diagram

V. POWER PROCESSING SYSTEM

Powering all of our electronics is essential for this project to work. This project requires a reliable power source that provides continuous power to supply the correct voltages and currents to the other system components. There are two aspects of power to this project, the power being provided and the power being processed, which is our power supply system. The power being provided is being supplied from a commercial power supply given to us by HP, which outputs 24VDC. Our power supply system can be broken into two sections: the low power exoskeleton and high power arm subsystems.

A. Exoskeleton Subsystem

The Exoskeleton power processing subsystem will provide continuous power to the exoskeleton components, which include the microcontrollers receiving sensory data, the encoders recording arm movement, the potentiometers recording finger movement, the gyro recording wrist movement, and Bluetooth module providing information via the microcontrollers to the Arm subsystem. Of the aforementioned subsystem components, all will be provided 5V power, with the gyro being the exception, which will be provided 3.3V. The total current carried by this subsystem will be less than 100mA, making this subsystem under 0.5W.

The Exoskeleton power subsystem will accomplish its goal using DC to DC linear regulators due to their simplicity to design. To efficiently provide power, and to reduce strain on components, three DC to DC regulators will be used. Instead of dropping the voltage from the given power supply to the voltage needed, the power will be distributed in stage to efficiently provide power. One DC to DC regulator to drop the voltage from the given 24V to 12V (strain relief stage), another to drop the voltage from 12V to 5V (5V output), and a third to drop the voltage from 5V to 3.3V (3.3V output). Fig. 9 shows the schematic for the exoskeleton power subsystem.

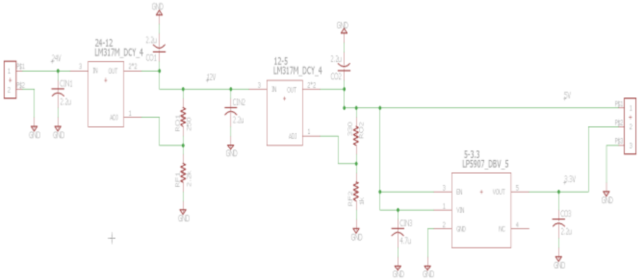


Fig. 9 Exoskeleton Power Schematic

Due to the nature of the exoskeleton being worn by a human user, all electronics needs to be compact and small enough to wear without putting a burden on the user. Designing and creating a Printed Circuit Board (PCB) is of critical importance as it allows an easy and light solution not having a breadboard be configured onto the exoskeleton, which compared to a PCB would be bulky and troublesome. One of the PCBs for the exoskeleton was created to house the power processing circuits, which is shown in Fig. 10. Vias were put in place around the perimeter of the board and one in the center of the board as the board does not use the bottom layer for signal tracing. This allows the bottom layer to be used solely as a ground plane for assistance in heat dissipation of the components, as linear DC to DC regulators can heat up easily.

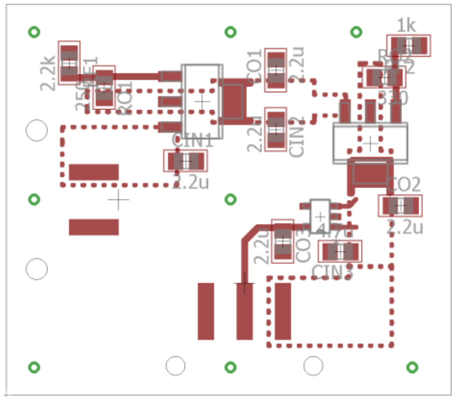


Fig. 10 Exoskeleton Power PCB

B. Arm Subsystem

The Arm power processing subsystem will provide continuous power the arm components, which will include the microcontrollers that are controlling the arm's movements via information received via Bluetooth, the Bluetooth module receiving information from the exoskeleton, and the fingers of the mechanical hand. Of the aforementioned components, the Bluetooth module and the microcontrollers will be fed 5V from the power supply at approximately 50mA, while the fingers will be fed 6V at approximately less than 1A each. The fingers are what classifies this subsystem as high power as the relative wattage of this subsystem is much higher than that of the exoskeleton approximately 30W.

The Arm power subsystem will accomplish its goal of providing so much power to the mechanical fingers and mechanical control using DC to DC linear regulators due to their simplicity; unlike the exoskeleton, this subsystem needs to take heat dissipation into much higher consideration, which will be discussed further. Much like the exoskeleton, the power processing will be done in stages to provide less strain on the components, with one stage dropping the provided 24V from the given power supply to 12V. Another stage will be used to drop the voltage from 12V to 5V to provide power to the embedded hardware, while a parallel stage will be used for the fingers, dropping the voltage from 12V to 6V. Fig. 11 shows the schematic for the arm power subsystem.

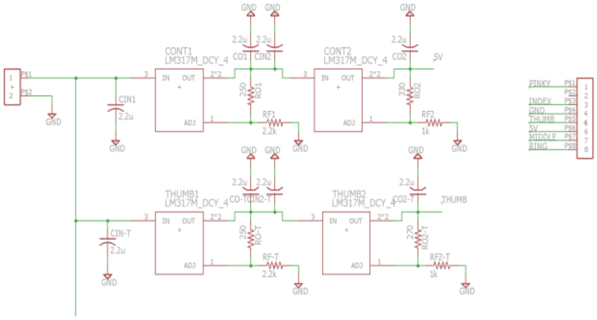


Fig. 11 Arm Power Schematic

When it comes to the design and creation of the PCB for the arm subsystem, heat dissipation is a heavy factor as servos with high loading requirements are being used. Providing 1A to each finger will prove to be a strain for the components which have to have the face the burden of the wattage being dissipation across it in the form of losses. To deal with this, large copper planes with the intent to help with heat dissipation were designed as per datasheet recommendations of each regulator used. These large power planes can be used when a heatsink is not ideal in the design process. Many thermal vias were created as well to assist in heat dissipation, as they help take the stress of the components by displacing the heat energy away from the component itself. Fig. 12 shows the PCB design for the Arm subsystem.

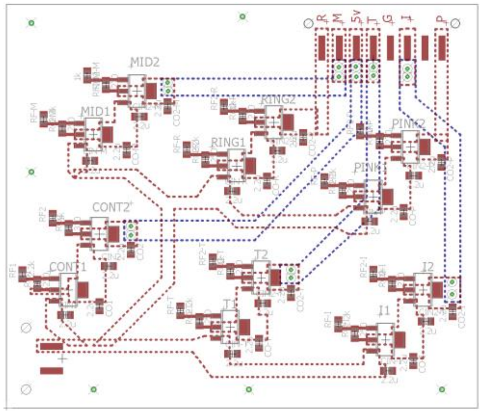


Fig. 12 Arm Power PCB

VI. SOFTWARE CONTROLS

Much like other areas of the project, the software controls are essential for this project to work. The software controls for this project mirrors the processor architecture. This means like the processors, the software is broken up into multiple parts, with different software subsystems in both the exoskeleton and arm systems. For each microcontroller on each PCB there is a separate program that needs to run. The software controls architecture deals with all of the following: intercommunication between MCU's, communication between arm and exoskeleton systems, motor controls for arm joints, and data manipulation for everything in between.

A. Exoskeleton Subsystem

For the controls board for the exoskeleton there are 3 processing microcontrollers. Two that deal with the encoders for the shoulder and elbow and one main MCU that handles all data inputs and then packages it up to be sent to the arm system. As mentioned in the processing section the reason for having an MCU dedicated to each encoder is due the fact that the MCU's only have 2 interrupt pins. These pins are needed as the MCU can't keep up with encoder without them. There are two encoders on the exoskeleton hence the two MCU's for them on the exoskeleton control board. The last MCU is the main MCU that takes in the digital gyroscopic data for the wrist and five analog potentiometer values for the fingers. It also receives data from the other two microcontrollers via I2C connections. For the interconnections of I2C, the main MCU is the Master as it receives the data from the two other MCU's. The other two MCU's are slaves to the main MCU. It then packages up all of the data into a 14-byte array and sends the data through Bluetooth using the RX and TX pins. Fig. 13 below shows the flow of how the software works.

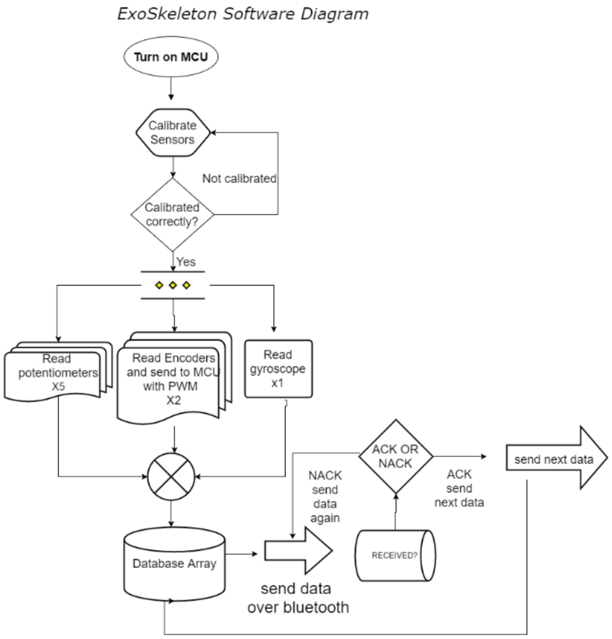


Fig. 13 Exoskeleton Control Software Flowchart

B. Arm Subsystem

Similar to the exoskeleton control board there are multiple MCU's on the arm control board. And as stated before for each MCU there is different piece of software that needs to be run. In this system there are 5 MCU's, one is the main MCU that receives the data from the exoskeleton system. The other 4 MCU's are servo subsystems for each joint, these include: shoulder, elbow, forearm and wrist. Each servo subsystem has an MCU dedicated it for the same reason as the control board, the MCU can only handle one encoder. This is more relevant for the Arm board as the encoders will be moving much faster on the arm than on the exoskeleton as they are attached to actual motors.

Each servo subsystem follows the basic architecture: first it calibrates motor and starts the position at zero. It then runs a PID algorithm. This is the key function that allows the motor to work as a servo. As the motor is attached to an encoder, it is possible to keep track of position, and with the algorithm, the user can tell the motor to go to a specific position with a certain speed. The other key function is the receiving function, this is used to get data from the main MCU. Once the PID algorithm starts it gets the data from the receive function and receives a destination value it must go to. It then calculates the error between the desired position and current position, this error is then multiplied by a certain gain Kp. This value is then used to calculate a PWM and direction value that is sent to the motor controllers that are driving the motors. One parameter to note is the window of values for the speed of the motors which is represented with a PWM value. The PWM values of 0-255 map to a speed of 0-100 percent for the motors. The motor controller takes the PWM value converts it to the speed percentage, and then outputs a corresponding voltage to the motor. The PWM window has been capped at 100 out of 255 due to the hardware limitations of the MCU. The interrupt pins can only be triggered so fast, so the speed is limited to allow for the MCU to keep up. To allow for the fastest response time, the values from the main MCU are sent to the servo subsystems every clock cycle.

Another item of concern is the resolution of the encoders and keeping this as high as possible throughout the entire architecture. To meet this requirement, the encoder value from the exoskeleton stays the same until it gets to the servo subsystem. Once it gets to this point, it is mapped with the values of -4000 to 4000 to -360 to 360 degrees. This is then mapped to the arm position, by using parameters for the motor. The motors are geared 100 to 1, this means for every revolution of each motor the joint of the arm moves 3.6 degrees. For instance, if the exoskeleton moves to 90 degrees or position 1000 for the encoder. The arm side would take the value of 90 and divide by 3.6 yielding the value of 25. This value is multiplied by 4000, one full revolution of the encoder to get to the value of 100,000 steps. The PID algorithm then sends the joint to this value yielding a position of 90 degrees.

Similar to the exoskeleton architecture for I2C. The main MCU is the Master, and all of the other MCU's are the slaves. The main MCU receives the data from the Bluetooth device through serial, it decodes the data package, then sends out the position to each joint. This also includes the hand, which is controlled by a different function but is also within the main MCU.

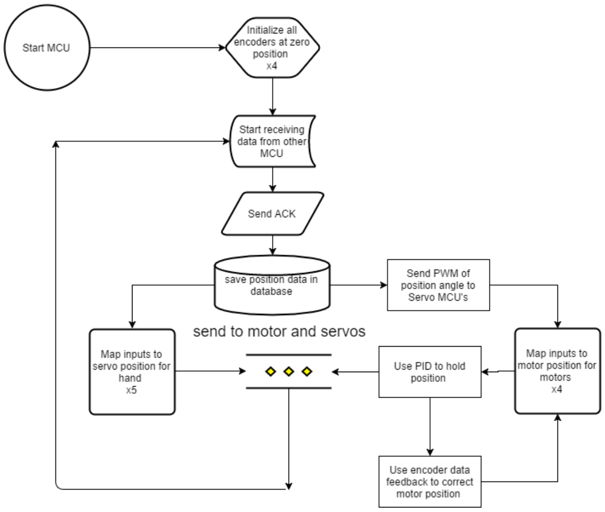


Fig. 14 Arm Control Software Flowchart

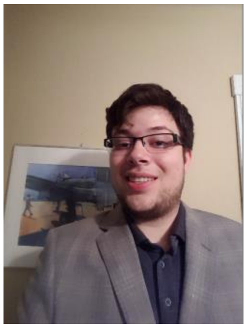
VII. CONCLUSION

As one can see the telerobotic arm system couldn't be realized without the all the aforementioned components discussed in this paper. The mechanical hardware, processing, communications, power systems, and software controls are integral parts to the project. The exoskeleton captures the user's motion through sensors, which is then sent to the robotic arm subsystem. The architecture was completely designed by the authors of the team. Through the knowledge gained from classes and internships the team was able meet the goals and requirement that were set by Hewlett Packard.

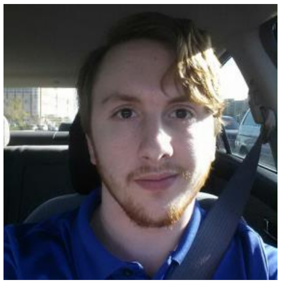
ACKNOWLEDGEMENT

The authors wish to acknowledge the generosity of Hewlett Packard of Boise Idaho for loaning the team the robotic arm for the semester. Specifically Hans Leidenfrost, Chris Morgan, and Rich Payne. All of their continued support made this project possible.

BIOGRAPHY



Carlos Cuesta will graduate with both a Bachelor's of Science in Computer and Electrical Engineering in December of 2016. Focusing on computer communication, software integration with its hardware counterpart, and embedded systems.



Devin Defond will graduate with a Bachelor's of Science in Electrical Engineering in December of 2016. He currently works as a Electrical Engineering CWEP at Lockheed Martin doing troubleshooting and testing. After graduation, he aspires to continue on with his education with a Master's in power electronics.

Akash Jinandra will graduate with a Bachelor's of Science in both Electrical and Computer Engineering in December of 2016. He is the chair of the IEEE undergraduate student branch at UCF. He has also done internships at Texas Instruments, Advanced Micro Devices, Precision Infinity, and Hewlett Packard. He hopes to get his Master's degree and MBA in the future once he's gotten more experience in the field.

Chang Ching Wu will graduate with a Bachelor's of Science in Electrical Engineering in December of 2016. He served as treasurer for the UCF student branch of IEE. He will focus his career in analog design, hardware integration, and robotic systems.