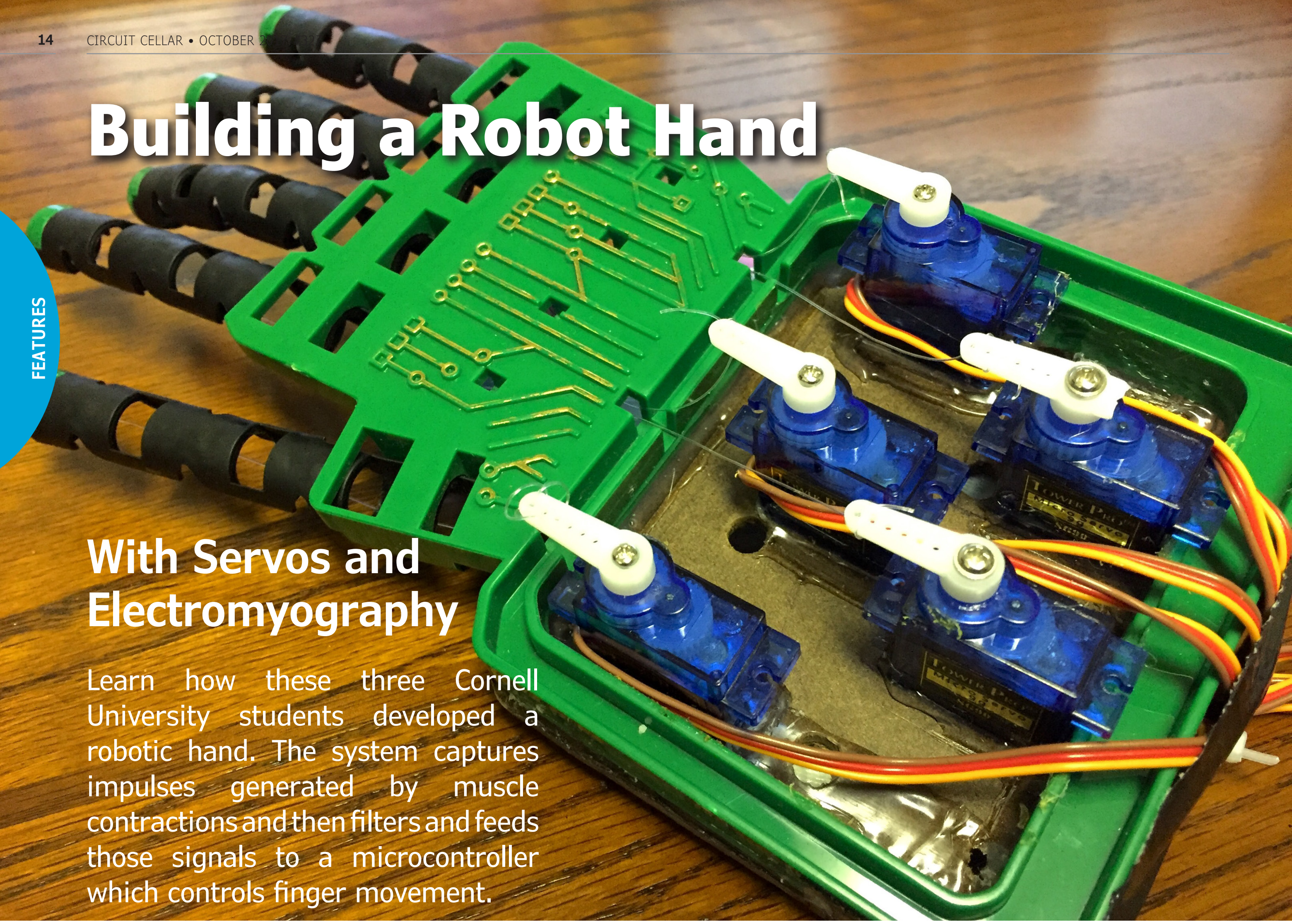

Building a Robot Hand

With Servos and Electromyography

Learn how these three Cornell University students developed a robotic hand. The system captures impulses generated by muscle contractions and then filters and feeds those signals to a microcontroller which controls finger movement.

PHOTO 1
Each finger is attached to a piece of fishing line which, when pulled, causes the finger to contract. The wires pulling each finger were attached to servos.

By
Michael Haidar, Jason Hwang, Srikrishnaa Vadivel

The inspiration for the project came from an interest in the biological signals in our bodies and how they control different body parts. As a person's fingers move, various muscles in the forearm contract. We wanted to analyze the signals sent to muscles in the arm and replicate the hand's motions onto a robotic hand. The design comprises a sensing and filtering system, and a motor output system. In its current form, all that is required for any user is an arm with intact forearm muscles.

The major benefit of this setup is that it is completely non-invasive, making it an attractive and simple option for all users. Additional development of this concept could help in creating non-invasive, advanced prosthetics. Furthermore, the input could be transmitted to a remote device which would replicate the user's actions. Possible applications of this could be performing remote surgeries, or remote control of humanoid robots.

SYSTEM OVERVIEW

Electrodes are placed on multiple areas of the wearer's arm to sense the electrical impulses generated by muscle contractions. The signal is then fed into a circuit which amplifies and filters it. The resulting output is sent into the Microchip PIC32 microcontroller which analyzes the readings. An algorithm decides which finger should be moved and a servo then moves the corresponding finger on the robotic hand.

To read the muscle contractions in the arm, surface electromyography (EMG) techniques are used. Electrodes are placed on two areas on the underside of the arm to read the electrical signals. These places were chosen through trial and error as they produced the greatest muscle contractions when the fingers were pressed. Since the electrodes are placed on the surface of the skin, the signal strength is very weak (around 2 mV at best) and noisy.

To improve the signal, the signal is fed into an instrumentation amplifier, followed by a high pass filter, a differential amplifier, and finally a low pass filter. The amplified and filtered signals are then fed into Analog to Digital Converters (ADC) on the Microchip PIC32 microcontroller. If certain conditions are met, a finger was determined to be pressed, and servos would pull a string attached to a finger of the robotic hand to contract it.

The base of the robotic hand is the '4M Robotic Hand Kit' **(Photo 1)**. Each finger is attached to a piece of fishing line which, when pulled, causes the finger to contract. The wires pulling each finger were attached to servos. In order to move a finger, the servos rotate 180°, pulling the wire and bringing the fingers from extended to contracted.

READING THE SIGNAL

A signal from the nervous system causes the muscle to fire. These signals are essentially a voltage difference across the cell membrane. An attenuated version of these voltage differences can be measured on the surface of the skin. The signals occur at varying frequencies depending on the strength of muscle contraction.

In order to differentiate the signals between the different fingers, we use two sets of two silver chloride electrodes on the forearm, all referenced to the same ground. For both sets, the two electrodes are placed about a quarter to a half an inch apart and affixed to the arm of the user with tape or foam sticking pads (**Photo 2**). When a nearby muscle contracts, it can be detected as a current of chloride ions on the skin. The chloride ions bind to a silver atom and "knock off" an electron (the reverse reaction happens as well). The electron can then travel through a wire connected to the electrode as a normal current.

To get a good signal, the electrodes had to be placed over the belly of the muscle. If the electrodes are placed too close to the tendons, the signal is inconsistent. Since there are several muscles in the forearm in close proximity to one another, it is difficult to isolate a signal from only one muscle. In order to differentiate between signals, we measured the same group of muscles in locations. The variable activity between these two inputs allowed us to discriminate between the contraction of different fingers.

PHOTO 2
Electrode placement: the ground electrode was placed on the wrist.

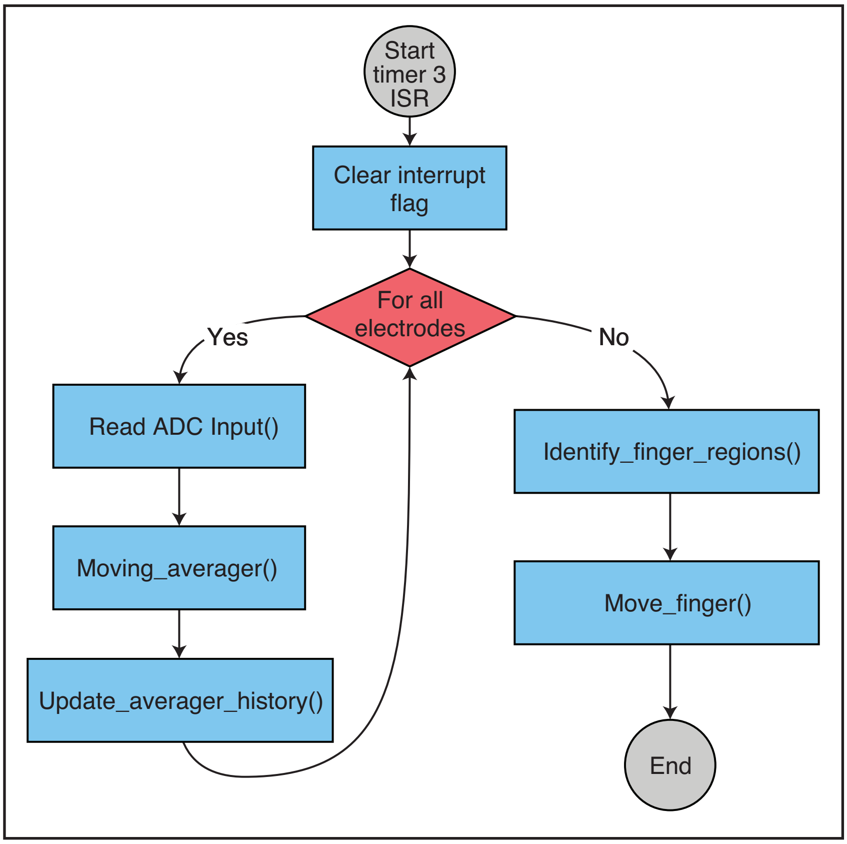

FIGURE 1
Logical structure for the software implementation

INSTRUMENTATION AMPLIFIER

The signal detectable on the surface of the skin is around 10 μV. In order to detect this signal with the microcontroller, we designed an instrumentation amplifier. Because the resistance of a human body is fairly high—on the order of megaohms—the input impedance of the amplifier needed to be significantly higher. Our instrumentation amplifier design is characterized by extremely high input impedance, low noise and high common mode rejection. This was ideal for our purposes because combating noise was our most significant challenge during this project. The basic design of the instrumentation amplifier was derived from the schematic in Medical Instrumentation: Application and Design [1], although the resistor and capacitor values are modified.

The amplifier contains two input op amps along with a two-stage differential amplifier. The gain of the first stage was 25, and the gain of the second stage was 214, for an overall gain of 5,350. The amplifier contains high and low-pass filters that are intended to reduce the signal received to the frequency ranges that were relevant to us. The low-pass filter has a cutoff frequency of approximately 1,500Hz, and the high-pass filter has a cutoff of 72 Hz. This was aimed at eliminating the 60 Hz line noise from light sources and nearby electronics.

SOFTWARE DESIGN

We read the input value, process the input, and move the servo based on the output of the detection algorithm. **Figure 1** shows the logical structure for the software implementation. The ADC has two functions: sampling & conversion. The sampling rate for the ADC is set by using a clock divider and changing the sample hold time for the ADC. All this is controlled in the initial configuration of the ADC. The ADC is configured to auto sample and convert. We also scan two analog input values (AN2 and AN3), from two different electrodes. The conversion results are placed in ADC buffer 0 and buffer 1. The ADC clock is around 588 kHz.

However, we do not get samples at 588 kHz because most EMG signals have a frequency range of 5-150 Hz. We use the Timer2 interrupt service routine (ISR) to read ADC buffer values at a predetermined sample rate. We experimented with different sampling rates and found that 500 Hz was good enough for the signal to not have any aliasing. In order to do this, we set the prescaler to 16, and loaded the timer with a value of 50,000. Most of the filtering is done in the ISR.

ABOUT THE AUTHOR

Michael Haidar is a senior at Cornell University studying electrical and computer engineering. Michael is passionate about embedded systems and biology. He is interested in neuroscience and its connection to electrical engineering

Jason Hwang is a senior at Cornell University studying electrical and computer engineering. Jason’s interest lies in the cross section of technology and business and how the two influence each other.

Srikrishnaa Vadivel is a senior at Cornell University studying electrical and computer engineering. Srikrishnaa is fascinated by device physics and its applications in processing biological signals.

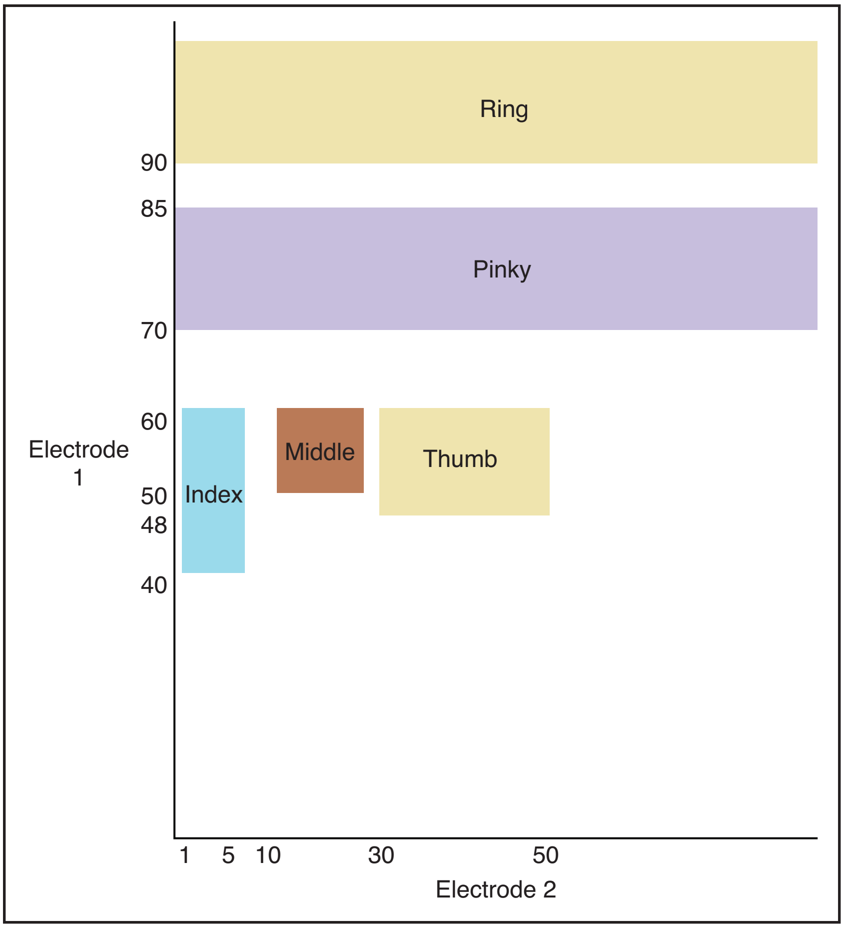

FIGURE 2
Finger regions based on electrode inputs

The purpose of the averager is to smoothen out the incoming ADC readings to produce a more consistent signal. Since our body signals contain many oscillations, the input reading to the ADC also contains many oscillations. By using a sliding window averager, the newest signal reading is averaged with the last 127 readings to produce a signal with less variations. The moving average filter uses fixed point arithmetic to save computation time.

Each set of electrode inputs has an ADC range of 0-1024. We empirically measure the block regions that each finger can correspond to. Even though the relative strengths of different fingers remain the same across different people or environments, the magnitudes themselves were people and environment dependent.

Noise plays a role in determining the movement of different fingers. In order to minimize noise, we made sure the wires were twisted together, with no connection to an AC power socket. Best results were achieved in an environment of low light, away from the electrical interference of fluorescent lighting. We then did empirical estimates of the range of ADC values for different finger contractions. The resulting "regions" for different fingers are seen in **Figure 2**.

Before contracting a robotic finger, we did additional checking for consistency in these threshold regions in order to prevent accidental movement. Our goal was to prevent unwanted fluctuation in output due to random noise. In our current program, we constrain it such that only one finger can be pressed at a time because multiple finger detection would require more electrodes. If this number of changes crosses an upper threshold for the finger, we mark that finger as being pressed. If this value goes below a lower threshold, we identify the finger as being released.

Each servo has a finite time to rotate by 180°. We empirically estimated this value to be 600 ms. We had to make sure that no other servo moved when another finger servo was moving, and make sure not to change the state of a particular finger unless 600 ms has passed. Both of these conditions were met by implementing a variable called `lock_motor`. This variable ensures that two finger servos do not move at the same time.

A Timer ISR was used to read the ADC buffer values at a fixed rate of 500 Hz. Since filtering was done at every ADC buffer read, we had to make sure that the computation time was less than 1/500 s = 2 ms. We used another timer to measure the time of execution for the whole filter and region detection algorithm. **Photo 3** shows the results isolating each of the five fingers. The internal 10-bit ADC was used in our design. This has 1,024 values with the highest value corresponding to 3.3 V. As a result, the resolution of the ADC is 3.2 mV. Moreover, since the ADC is not accurately calibrated, this range was reduced and hence we had a lower voltage range to work with. However, this was not a serious limitation given that our input signals were on the order of a few hundred millivolts.

Surface EMG signals have a bandwidth of 5-500 Hz. However, the most prominent signals occur in the range of 5-150 Hz, with only a few signals exceeding this range. Our hardware design accommodates a major part of this frequency range. The sampling rate is 500 Hz, which gives a Nyquist frequency of 250 Hz. The most prominent frequencies are still within this range. Therefore, the bandwidth constraint was satisfied.

NOISE SOURCES

Noise sources are major deterrents in using surface EMG. We tried to minimize or eliminate noise sources where possible without using extreme measures. The dominant noise source is the 60 Hz line noise. We eliminated this by using an isolated power supply. Line noise can also be introduced on the body when there are power sockets in the vicinity. This was partially remedied by using the high pass filter. Another common source of noise is from fluorescent light bulbs. These oscillate at twice the line frequency, and all odd harmonics of this noise can be induced on the human body. We just tried to reduce this by testing in places away from fluorescent bulbs.

Another source of noise that we observed is the interference from other muscles in close proximity, such as other muscles in the hand, or even muscles farther away, such as those in the chest. This can be reduced by stretching the hand far away from the body. Magnetic coupling that is induced in the wires from external sources should also be considered. Any current carrying wire loop has an inductance associated with it. This causes magnetic field coupling that induces some noise current in the circuit. This is reduced by using twisting the wires together and making sure the length of the wires is not

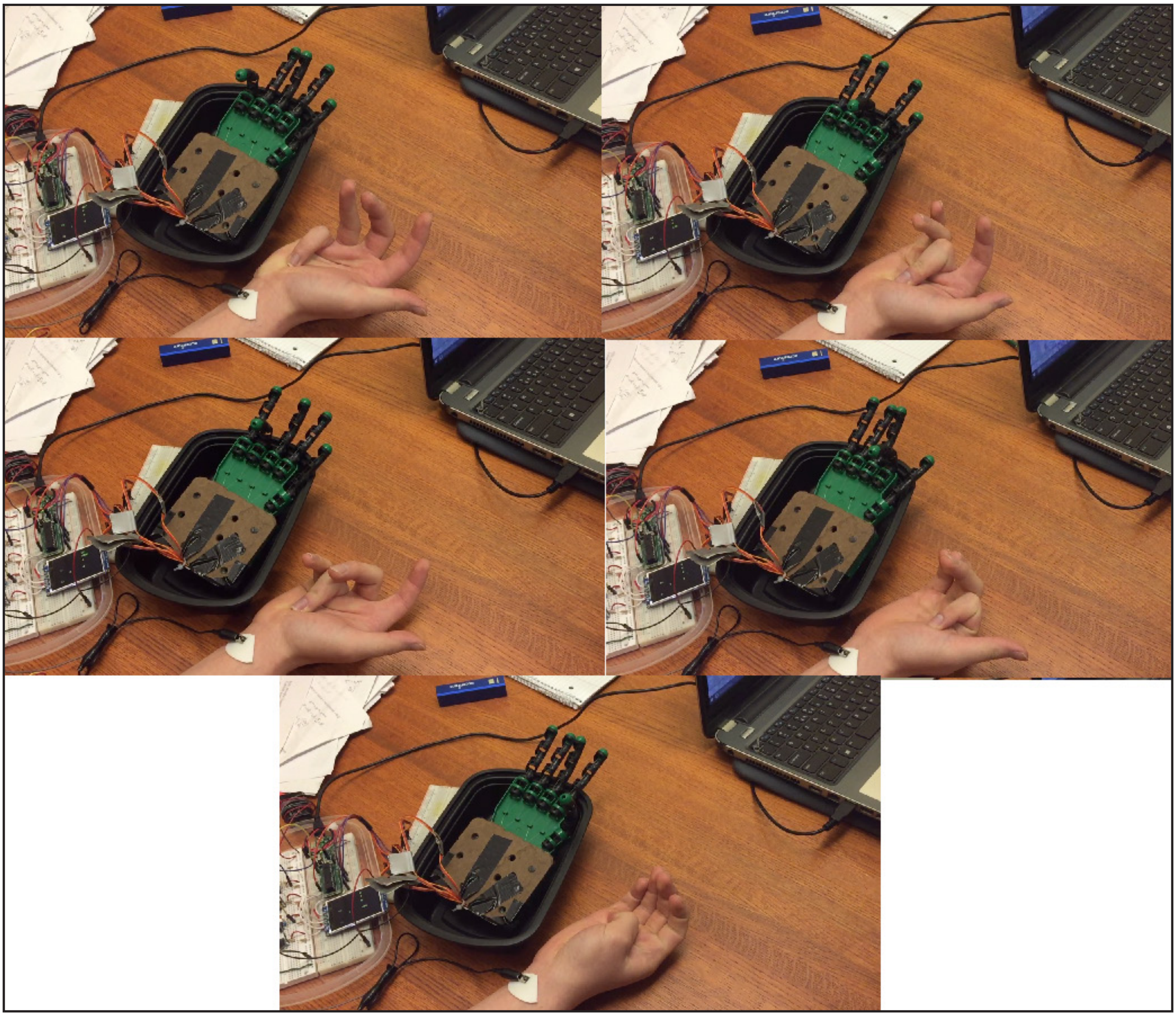

PHOTO 3
Show here are the results—isolating each of the five fingers.

too long. This ensures that the inductance of the wires is at a minimum, and hence the magnetic coupling is at a minimum.

SAFETY CONSIDERATIONS

In order to enforce safety, we designed the entire circuit to be completely isolated from any 110 V outlets. Theoretically, if there had been a pathway to 110 V ground somewhere in the circuit and the test subject were to somehow touch the hot wire of an outlet, he or she could be seriously injured or killed. Battery power was used to ensure that the power supply of the circuit was isolated.

Another possible issue could arise if the circuit were connected to two different outlets with different grounds. If the test subject is connected to electrical equipment from one outlet and some other equipment from another outlet which is referenced to a different ground, this could cause there to be a voltage across the test subject. If the voltage is large enough—or if the subject's heart is weak enough—this could potentially cause the heart's pacemaker to become irregular, leading to cardiac arrest.

Isolating the entire circuit eliminates these issues. In order to test the circuit safely we wrote a simple program for an Arduino to be used as an oscilloscope. Connecting the circuit to a normal oscilloscope would have created a path to ground via the oscilloscope's power cord. We were then able to use a laptop running on battery power—again to isolate from ground—to measure signals.

A secondary issue we discovered was that the conductive gel that we applied to the electrodes would occasionally get in between the electrodes and create a short, leading to the user feeling a slight tingling sensation. In order to eliminate this issue, we applied the gel more judiciously and made sure that the space between electrodes was clean and dry at all times.

CONCLUSION

Our design met our expectations and goals set at the beginning of the project. Our goal was to use solely electrode inputs to mimic on a robotic hand the movements of our own hand. **Table 1** shows a cost breakdown of the project. At the end of the project, we were able to isolate the signals for each finger and move all five fingers independently of each other on the robotic hand. Judging the design of our circuits, we were able to provide enough gain in the amplifier circuit for us to convert the signal from a few μV in amplitude up to one that is a few volts large and readable by the ADC. We were also fairly successful in filtering out unwanted noise so that the signal we received was more accurate.

Item	Cost
Robotic Hand	$10.99
Electrodes	5 x $1.11
Servo Motors	$11.28
PIC32 microcontroller	$5.00
Microstick	$10
Breadboards	3 x $6.00
Jumper cables	4 x $0.10
TFT LCD	$10
9 V batteries	2 x $2.00
Total Cost:	$75.22

TABLE 1
Cost breakdown of the project

We could potentially add extra functionality to the hand such as the ability to sense multiple fingers being pressed at the same time. Another way to improve the project would be to improve the responsiveness of the fingers and have them move with little delay when compared to our own hand. This may mean playing around with the thresholds and increment/decrement values so that the fingers move more quickly. We believe there is room to improve our project and add more advanced capabilities. For example, using a more robust and custom made robotic hand with gripping capabilities, or using a neural network to improve the algorithm's ability to adapt to widely varying environments and users.

circuitcellar.com/ccmaterials

REFERENCES

[1] Webster, John G., and John W. Clark. Medical Instrumentation: Application and Design. Hoboken, NJ: John Wiley & Sons, 2010. Print.
[2] Microchip PIC32 Datasheet: ww1.microchip.com/downloads/en/DeviceDoc/61156H.pdf
[3] SG90 Tower-pro Servo Datasheet: www.micropik.com/PDF/SG90Servo.pdf