Building a Cheap Prosthetic Hand with Mini Computers

Jeron Osguthorpe

SUU SUCCESS Academy

Introduction

In my 2021 SUCCESS Science Fair Project, I am making a Cheap Prosthetic Hand with a Micro:Bit and Raspberry Pi. The Raspberry Pi is a minicomputer that can be coded to do almost all functions that could be put on it. The Mirco:Bit contains a lot of sensors that will send the serial of the sensors to the Raspberry Pi. When the Raspberry Pi receives these sensor messages it will run some motors that will then control the hand. The hand will be a 3D printed hand. This makes it so that everything in the project is cheap.

Project

- Engineering Goal: Build a Cheap Prosthetic Hand using Mini Computers
- Hypothesis: If I can Build a Cheap Prosthetic Hand, Then it could help low-income people afford prosthetic devices.
- Expected Outcomes: I can build a prosthetic hand using a 3D printer, and two mini-computers which will become an efficient way to have a prosthetic hand.

Procedures

1. Use the makecode.microbit.org editor to code the program for the microbit.
2. In the MakeCode Editor Create an “on” statement for every sensor on the micro:bit (shake, logo up, logo down, screen up, screen down, tilt left, tilt right, free fall, 3g, 6g 8g)
3. Write the serial code for each “on” statement to say “serial write line ‘(sensor)’” or “serial write number ‘(the number you can assign for each sensor)’”
4. Test the Serial Code with the Online Emulator making sure the serial output is correct
5. Save your project to your MakeCode Profile (if you choose to make one) or to your local computer to save keeping.
6. Download the .hex file.

7. Plugin your Micro:Bit to your computer

8. Copy and Paste the .hex file from earlier into the new partition titled “MICROBIT”

9. Get an 8gb or higher MicroSD Card as Well as a Raspberry Pi (in our case it is the raspberry pi 3B+)

10. Plug the MicroSD card into your computer

11. Go to the Official Raspberry Pi download site and download the “Raspberry Pi OS With Recommended Software” as a .zip folder (DO NOT EXTRACT)

12. Go to the BelenaEtcher Website and Download/Install the app

13. Open BalenaEtcher

14. Under “Device” Select your MicroSD card

15. Under “File” select the .zip folder that you downloaded earlier

16. Press Flash To Device

17. Once the Flash Is Completed remove the MicroSD Card from the Computer and put it into your Raspberry Pi

18. Plug a Monitor into your Raspberry Pi with either the Display or HDMI Ports

19. Plug a Mouse and Keyboard into the Raspberry Pi

20. Supply Power to the Raspberry Pi and wait for Raspberry Pi OS (formally Raspbian) to open

21. Once In, Finish Setting your Raspberry Pi and Reboot

22. Once back in, plug your coded microbit into your raspberry pi via USB

23. Open a new terminal window

24. Type in “ dmesg | tail “ to find out what “ /dev/ “ node the microbit was assigned (e.g. /dev/ttyACM0).

25. Type “picocom /dev/ACM0 115200 “, replacing the number with the number you found in the previous step, if no number given use “ACM0”. If error install the “screen” program (sudo apt-get install picocom)

26. You should now see your serial data from the microbit

27. Make a new .txt file, this is where we will write the code to run the command

28. Write the Code in Python for the raspberry pi connecting to the micro:bit. Using sources from Github from people over in Latin America

29. Talk to a orthopedic doctor for how hands move

30. Design, and print a 3D hand. (or use this one https://3dprint.nih.gov/discover/3dpx-000524) OR make your own using AutoCAD or TinkerCAD

31. Buy motors for the 3D hand and install them

32. Write the python program for “if” statements for the motors when they moved based on the results the orthopedic doctor gave you.

33. Connect the Raspberry Pi to the Micro:Bit and to the motors. Then connect the power to the raspberry pi which will supply power to the micro:bit

34. Connect the power to the motors

35. Begin the testing of the hand.

36. Write down the changes that are made when you move the hand.

37. Change the code according to what works and what fails.

38. Give final testing

Results

In the end, my project kinda worked. When finishing up the code on the 31st of December of 2020 I was able to get the hand to start moving when I switched the fishing line to

some wire that was able to withstand some more pressure of the motors, I also had to use a special UV glue, as well as super glue, as well as hot glue, as well as rubber bands to keep all the fingers together, this in fact is something I could change using a more sophisticated system from the Institute that was stated in the materials and methods section. This would make the project significantly stronger allowing me to have more pressure with the motors without breaking the project apart, which ended up happening multiple times before I used the UV glue and the other types of fasteners. As an end result, I was able to slowly get the fingers moving, one by one until I was able to get them all to move and I was actually able to get the hand to close and open. In the end, my engineering goal was accomplished and I was able to make a prosthetic hand that would only cause $100-200. As part of the computer results that came the computer returned this code. Each column is a different time, I used different columns to make it take up less page space.

IH1 / GPIO Pt. 1	IH2 / GPIO Pt. 2	IH3 / GPIO Pt. 3
Step 1 - 525	Step 1 - 525	Step 1 - 525

Discussion

During the working of my project, this made me have to do many hours of work, in fact, according to my timing I achieved over 45 hours of work. During this 45 hours of work, I first went and talked to a professional prosthetic doctor and talked to him about designs, he advised me NOT to look at designs at first because we don't want me to make a design that is based off another design that someone else had made. After all of this I used a program called TinkerCAD and was able to make a design for my hand, it looks like a Minecraft hand to be honest, but in the end it was able to be printed at my good friends school, LAUNCH. They were willing to allow

me to print my hand off using one of the Makerbot Pros, which is a 3d printer. After this, I struggled to break the bases off of the hand and the fingers and ended up having to use a hot knife, and a very precise saw to cut off the rest of the bases as well as a hammer. After this I was able to put the hand together, although, I wasn't able to find some screws small enough to use, so I decided to settle on using hot glue, this ended up not working because it fell apart many times and I ended up resorting to using a superglue/UV glue mix as well as a little bit of hot glue and some rubber bands.

I then had to make the arm, and my original design for the arm was very big, and so instead of using my giant Solderless Breadboard from radioshack, I resorted to using a smaller board from RadioShack to run the 2003j Stepper Motors and driver boards that I had bought off of amazon.com. I then used female-female, female-male, and male-male connectors to connect the raspberry pi to all the motors. I also used legos and some special 3M sticky tape to keep everything onto a lego platform that could act as part of the arm. I then was able to use some screws and connectors to connect the motors to the hand itself and then after putting the strings on, I was able to use popsicle sticks to put the hand and the arm together and to secure the Micro:Bit to the arm. I was then able to start the rest of the code for the project, and using a bunch of sources from Github.io that people had made open to the public, I was able to officially make connections to the driver boards. After all of this, it was time for testing.

I ran into many errors while testing, some of them having to do with the code but most of them having to do with the hand itself. The string on the hand ended up snapping many times and I ended up actually switching to a very flexible wire that was supposed to handle enough force, at the end I was able to get the hand to actually work, one finger at a time, because the Raspberry Pi was only able to run one python string at the same time. In the end, one of the

motors decided to not stop and I had to hold the finger so it didn't break and ended up snapping the entire wire itself.

This project could definitely be changed for the better, there are many things that could make the project run a lot more smoother. I would be able to use the 3d files from the Institute to make the project more stable and make things work better. It also is a string based format so it would work too. I could also be able to make this more mobile by using a battery pack to run the raspberry pi and getting a 3in touchscreen for the Raspberry pi to run the small programs, this actually would only add another about $20 to the project still staying in the very cheap range for when it comes to prosthetic devices. This would also be rechargeable and updatable allowing the project to be altered and improved with little to no cost.

Conclusion

At the end of the project, the project could be both considered a success and a failure. There are many things that went well and many things that went wrong and many things could be improved and I could make it more easily accessible and cheaper. I would be able to finish this project again next year or do it again on my own time. This would allow me to make many improvements, both with code and with sensors allowing me to possibly have this be a real usable project in the future.

Acknowledgements

I would like to give a thanks to Dr. Matthew Bracken, CPO from Utah Prosthetics and Orthotics in Cedar City, UT. He was willing to sit and talk with me for about an hour, helping me with this project. I would also like to thank the people who made the github project files that I ended up basing my project around. I would also like to thank Shanae Osguthorpe, who was

willing to spend the literal entire day of December 31, 2020 with me working on this project, we literally worked from 9AM to 11:30 PM that day.

Risks and Safety

The risks are the possibility of headaches, fatigue, and eye strain from staring at a computer coding and fulfilling commands with the computer.

Data Analysis

I can collect information and data from both the Micro:Bit/Raspberry Pi connection and by watching the hand/arm itself to see how it moves and record it in a journal. I will then try to put them in a graph or in a list of information using either Google Docs or Google Sheets.

Bibliography

Cash, J. C. (2019, February 7). *Exporting micro:bit sensor data*. Make:Learn. https://makelearn.org/2019/02/07/exporting-microbit-sensor-data/

micro:bit Help & Support. (2020, January 6). Micro:Bit. https://support.microbit.org/support/solutions/articles/19000022103-outputing-serial-data-from-the-micro-bit-to-a-computer

McKay, D. M. (2019, July 1). *How to List Your Computer's Devices From the Linux Terminal*. How-To Geek. https://www.howtogeek.com/426199/how-to-list-your-computers-devices-from-the-linux-terminal/

Gestures — BBC micro:bit MicroPython 1.0.1 documentation. (n.d.). BBC Micro:Bit MicroPython. Retrieved September 30, 2020, from https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/gestures.html

Osguthorpe, J. O., & Adam, A. (2020, September 18). *How do I make an "if" statement with serial data?* Stack Overflow. https://stackoverflow.com/questions/63948030/how-do-i-make-an-if-statement-with-serial-data

Anderson, R. A. (2020, August 19). *remihacker5/arduino-usb-controller*. GitHub. https://github.com/remihacker5/arduino-usb-controller

J, Z., D, K., J, P., J, S., M, P., A, C., & C, F. (2015, January 20). Cyborg Beast (Original Design). Retrieved October 01, 2020, from https://3dprint.nih.gov/discover/3dpx-000524

https://youtu.be/Dc16mKFA7Fo

(n.d.). Retrieved January 15, 2021, from https://projects.raspberrypi.org/en/projects/physical-computing/14

(n.d.). Retrieved January 15, 2021, from https://www.raspberrypi.org/forums/viewtopic.php?t=137714

Barnett, J. (2014, April 11). Controlling DC Motors Using Python With a Raspberry Pi. Retrieved January 15, 2021, from https://business.tutsplus.com/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051

Clinton. (2018, November 23). How To Control A Motor with the Raspberry Pi - Tutorial. Retrieved January 15, 2021, from https://core-electronics.com.au/tutorials/how-to-control-a-motor-with-the-raspberry-pi.html

Custom-Build-Robots. (2017, June 13). Custom-build-robots/Stepper-motor-28BYJ-48-Raspberry-Pi. Retrieved January 15, 2021, from https://github.com/custom-build-robots/Stepper-motor-28BYJ-48-Raspberry-Pi/blob/master/decision-maker.py

GPIO in Python. (n.d.). Retrieved January 15, 2021, from https://www.raspberrypi.org/documentation/usage/gpio/python/README.md

GPIO. (n.d.). Retrieved January 15, 2021, from https://www.raspberrypi.org/documentation/usage/gpio/

Peppe8o. (2019, November 13). Controlling a stepper motor with Raspberry Pi Zero W. Retrieved January 15, 2021, from https://peppe8o.com/control-a-simple-stepper-motor-with-a-raspberry-pi/

Stepper motor 28BYJ-48 – ULN2003A controller – Raspberry Pi and Python. (2017, June 14). Retrieved January 15, 2021, from https://custom-build-robots.com/electronic/stepper-motor-28byj-48-uln2003a-controller-raspberry-pi-and-python/8862?lang=en