Stick for the visually impaired

Mihnea Tîrcă

January 16, 2023

Contents

1 Introduction

The purpose of this project is to improve the simple white cane used by blind people by attaching an ultrasonic sensor to it that measures distance.

I chose this project because it occured to me that never have I heard of a product on the market for blind people that uses a sensor for measuring distance to obstacles. Also, rather than the user relying on only the sensor for getting around - and being more prone to error, I chose to attach it to the existing product so that the blind person has multiple ways of getting notified of obstacles.

While I did come up with the idea myself, I quickly found that I am unsurprisingly not the only person who thought of making such a prototype. Other people's projects didn't influence my project too much: I already had the idea of how to implement it. Nevertheless, there were a couple of minor implementation details that I borrowed from other projects, which I will be discussing later in this document.

2 Proposed solution and implementation

Components needed:

- Battery
- Vibrating motor
- Microcontroller board
- Ultrasonic sensor

The specific components used in this project are:

- 9V Battery
- 9V Battery Case
- Arduino Nano
- 1027 Disc Vibration Motor
- HC-SR04 Ultrasonic Distance Sensor

The solution I've come up with is attaching the components to a standard cane for the visually impaired. I used an ultrasonic sensor for getting the distance to an obstacle, and by attaching a vibration motor on the cane handle, the user would hold the stick with his thumb on the motor and would be notified of obstacles. The vibration motor is attached to a PWM pin on the board, which allowed me to increase the vibration the closer the obstacle is. I attached the ultrasonic sensor near the bottom of the stick, so that it "sees" obstacles such as a raise in the ground level because of a sidewalk. The Arduino Nano and the Battery can be attached anywhere on the stick. I chose to attach the board somewhere in the middle of the stick, and the battery a little closer to the top in order for the user to be able to turn the battery on and off easier (the battery case has a switch), as well as for avoiding adding more weight to the bottom of the stick, which would make holding the stick and swinging it side by side harder than if the center of mass is higher. With all this in mind, I designed the circuit as in Figure 1.

2.1 Circuit connections

Firstly, the battery needs the + and - to be connected to the V_{in} and GND, respectively. The vibrating motor needs to be connected to GND and a PWM pin, such as D3. The ultrasonic sensor needs the V_{cc} and Gnd to be connected to 5V and GND, respectively. The Trigger pin of the sensor emis the frequency, while the Echo pin captures it. When trying to connect the two to the Arduino Nano pins, I found unconclusive information online about whether they need to be connected to PWM pins or not. Some said you should be connecting the Trigger pin to a PWM, while others said you should be connecting the Echo pin to a PWM. Thus, I connected both to PWM pins: Trigger to D6 and Echo to D5.

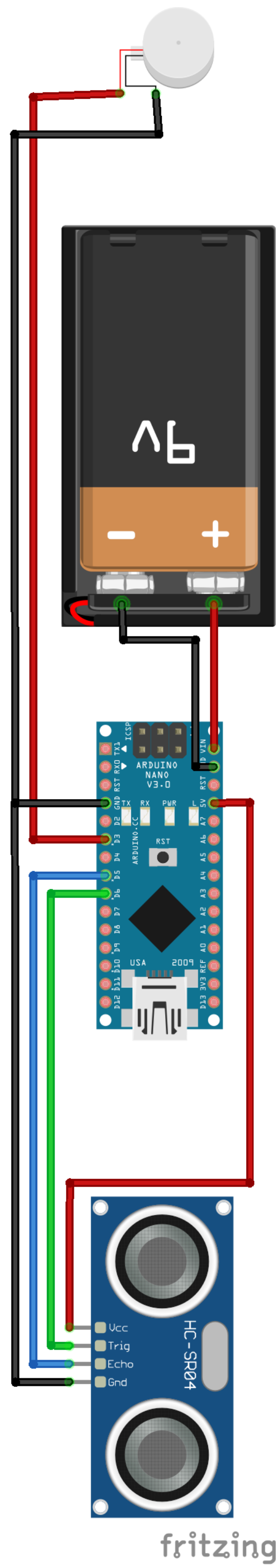

Figure 1: Circuit

2.2 Implementing the circuit

This circuit is simple on paper, and while it can be argued it is simple to implement it on the stick as well, it was not as quick. I fixed the sensor on the stick by ductaping the end of wires connected to it. The wires needed to be long in order to travel half-way up the stick. The only wires I found were some that were too long so I wrapped them around the stick. The battery holder I also ductaped. I folded the ductape into itself and stuck it with glue in order to make a double sided tape, and stuck the battery holder to it. That was not enough, so I also wrapped ductape around the battery holder and the stick. The vibrator, I stuck to the stick with double sided tape in much the same way, and also glued it to the tape. This is what I think caused the vibrating motor to break, as it sometimes stops working or emits high pitched sounds.

The biggest problems I had were sticking the board onto the stick, as the Arduino Nano has pins sticking out of it. I did not want the pins to be sticking towards the outside of the cane, as there's the risk of them being hit. Trying to fix it with its pins towards the stick was also not possible as the stick was not narrower than the distance between pins, so the pins would be sticking in the cane. So, I searched the internet for projects like mine to see the solutions other people came up with[1], when I found one person using a styrofoam block to stick the board on. So, I used the styrofoam block I got with the board, made a final decision on which pins I want to use, and cut the styrofoam in order to make space for the wires to go into specific pins, while the rest of the pins were safe in the styrofoam.

Making the connections was the longest part of making this project. I needed a lot of female-to-female wires, and I only had female-to-male. So, a lot of the times I cut the wires on the male end, used a lighter to expose the wires, then glued and ductaped them together to make female-to-female wires. I also ductaped a lot of wires to the stick itself in order for them not to stick out too much.

2.3 Programming

My initial thoughts were to simply get the distance, and using the "rule of three", getting the PWM output to the vibration motor:

$$400cm..........0PWM$$

$$0cm............255PWM$$

$$distance........?PWM$$

Where 400cm is the maximum distance at which the ultrasonic sensor can see, and 255 is the maximum value that can be written on a PWM pin. That would net us the formula:

$$PWM = 255 * (1 - (distance/400)$$

During testing I realized that I might not want the sensor to see at its maximum distance. For example, if the user were to use it indoors, the motor would always be vibrating. I also tinkered with the idea of not having 255 be the maximum PWM value, as the vibrations might be too high and and be uncomfortable for the user. Thus, I had changed the formula to:

$$PWM = MAX\_PWM * (1 - distance/MAX\_DISTANCE)$$

As a side note, later in my testing, I also defined a variable for the minimum PWM, for I noticed that in the approximate interval [0,50] PWM, the motor would not vibrate, so I defined 50 to be the MIN_PWM.

There were other problems that I encountered: the ultrasonic sensor I used is not very accurate, and so the readings were very volatile. The distance read would jump a lot, without moving the stick.

[1] I found the project on https://projecthub.arduino.cc/, but it seems the author took it down in the meantime, so I cannot credit him

Thus, I changed my code to get a number of consecutive readings - let's say 10 - and then get the average distance of the readings. This way, the discrepancy between readings would be alleviated.

This solution made things better, but I noticed the sensor would return a reading of a distance lower than it actually is. I came up with the solution of, after the 10 consecutive readings, getting the highest one, instead of the average distance.

After more testing, I noticed that the motor would (correctly) vibrate a little bit too much even when at a respectable distance from the obstacle. The linear formula for getting the PWM was not satisfactoy, so I wanted to change it to an exponential one. The solution boiled down to getting an exponential function between two points: (MIN_PWM, MIN_DISTANCE=0) and (MAX_PWM, MAX_DISTANCE):

Figure 2: Exponential function

The exponential function has the form

$$f(x) = ab^x \tag{1}$$

and the two points are $(0, PWM_{max})$ and $(distance_{max}, PWM_{min})$
Using a, we substitute PWM_{max} in the equation to get b

$$a = PWM_{max} \tag{2}$$

$$\implies f(x) = PWM_{max}b^x$$

We know that $f(distance_{max}) = PWM_{min}$ so we get

$$PWM_{min} = PWM_{max}b^{distance_{max}}$$

From this equation we get b

$$b = \sqrt[distance_{max}]{\frac{PWM_{min}}{PWM_{max}}} \tag{3}$$

Now that we know a and b, we can substitute them in $f(x) = ab^x$

$$(1), (2), (3) \implies PWM = PWM_{max} \cdot \frac{PWM_{min}}{PWM_{max}}^{\frac{distance}{distance_{max}}}$$

This gives, in my opinion, better results when using the stick.

3 Conclusion

In conclusion, the project went through a couple of changes, and in the end I think it turned out well for a prototype. It could, of course, be improved. The vibrating motor broke when glueing it to the stick so perhaps a flexible material between the motor and the stick could be placed - such as styrofoam, so that it would absorb the vibrations. While I tried to keep the wiring tidy, they are still a bit all over the place. The components and wires could be placed inside the stick, as some white canes for the visually impaired seem to be hollow. Also, there is another improvement that could be made. The cane could have a setting for the maximum distance at which the sensor can see. The reason for this is that the maximum distance should be lower indoors and higher outdoors. Outdoors, the user might want to be notified of an obstacle that's one meter away, while indoors that might make the stick vibrate constantly.

Nonetheless, I am satisfied with how the project turned out, especially with my solutions, like making the vibrations exponential so it wouldn't vibrate as hard when relatively far away, or getting multiple consecutive readings to process the final one. While it is a prototype, it could lay the groundwork for an upgraded project.