**Proxicane: a concept of a low cost, open source, wearable, obstacle avoidance system for blind/partially sighted users. (text first then diagrams/images)**

[explain what your project is – explain how blind people use canes/ why your solution is better]

[replacement of the cane]

Normally, blind/partially sighted person would use a cane to avoid obstacles while navigating a built environment. Depending on the level of impairment different types of canes are used. All are visible and require the user to constantly move the cane as they walk. Proxicane, is an alternative to using the cane, it uses haptic feedback and ultrasound sensors to alert the user when they are near a obstacle, lowering the cognitive load of the user, thus the user is more able to focus on their environment. Proxicane, although a concept, if mass manufactured and reengineered commercially could provide a blind user the independence to navigate a built environment without a need to actively interact with the built environment, they could move discretely and confidently.

(<https://www.rnib.org.uk/cane-explained> show diagrams of current cane/compare pictures to yours)

(insert labelled picture of finished Proxicane)

A similar project that has inspired me is an echolocation device mounted to the arm rather than the leg <https://challenges.openideo.com/challenge/refugee-education/ideas/the-world-s-first-fab-lab-in-a-refugee-camp/comments>

**How does Proxicane Work?**

An 9v battery powered Arduino has a ultrasound sensor connected to it and a detachable coin cell sized vibration motor. Both the vibration motor has a Velcro fastening and a Velcro strap, the Arduino also has its own Velcro strap. This consists of one Proxicane unit.

For each leg, only 2 Proxicane units are required. One unit faces forward, and another unit will face 90 degrees to the side of the user.

[show in diagram – makes it clearer]

[labelled diagrams showing Proxicane near a wall- vibrating]

When a user moves a set distance near an obstacle (which is any hard/soft surface) the Arduino turns the vibration motor on, which vibrates letting the user know that they are near an obstacle and if they move any further to it, they may collide.

The Arduino is constantly measuring the surrounding distance via the connected ultrasound sensor.

If the user moves more than a set distance (adjustable by the user) from an obstacle the Arduino switches off the vibration motor, thus the user knows they are not going to collide.

The detection distance is currently set at 30cm but this can easily be modified by plugging the Arduino into any computer via a USB cable and changing the *collision\_distance* variable, the program code is provided and is fully open-source.

**What is ultrasound and how does the ultrasound sensor work?**

[explain labelled diagram of ultrasound sensor]

[replicate diagram here: <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>]

Ultrasound is sound wave frequencies that are above 20,000 Hz which is the highest frequency audible to the human ear. Just like sound waves, ultrasound can travel through gas, liquids and solids but not vacuums. Most importantly Ultrasound waves travel in a very straight line (longitudinal). When ultrasound waves hit the surface of an object they are reflected back, an ultrasonic sensor can detect the reflected wave.

Since ultrasound travels at a speed of roughly 340m/s through air (at room temperature), the distance can be worked out by using the speed formula.

[insert speed diagram]

Therefore, *Distance(m) = speed(m/s) × time(s)*

An ultrasound sensor such as the HC–SR04 (<https://www.piborg.org/sensors-1136/hc-sr04>) can using its two speakers which act as a transmitter and receiver calculate the time it takes for a single pulse of ultrasound waves to be transmitted and received. With time calculated by the sensor (the sensor has onboard circuits that are designed for this), the distance can therefore be calculated by the Arduino.

**What is the coin cell vibration motor and how does it work?**

The coin cell vibration motor used in Proxicane is the same type of low-cost vibration motor normally found in modern smartphones.

Coin cell vibration motors are miniature DC motors that spin a unbalanced weight. Spinning the weight creates vibrations that can be felt by the user. This one of the cheapest methods to achieve haptic feedback.

[insert diagram explaining this]

[<https://www.precisionmicrodrives.com/vibration-motors/coin-vibration-motors/>][explains in high detail how a coin vibration motor works]

**Why was haptic feedback used instead of sound or other outputs**?

[explain what is haptic feedback]

Sound was considered as a feedback to the user if they were near collision, however there is an increased cognitive demand for the user because the user as well as using their hearing to already navigate their environment they would also have to focus on hearing a specific sound from the Proxicane device. By using haptic feedback instead, the user’s is free focus all of the their hearing on navigating a built environment (for example listening for nearby cars).

Also, the sound would make Proxicane less discreet for the user, it would draw unnecessary attention from passer-by, which would reduce the user’s confidence in navigating a built environment.

Moreover, a built environment is normally loud, the sound output of the Proxicane would have to be loud just for the user to hear it outdoors but then a serious problem arises if the user enters a quiet indoor environment where the loud sound from the Proxicane would not be appropriate.

Moreover, the user in any built environment will come across many obstacles at a regular rate, a constant repeating sound to the user may be will become a nuisance to the user.

Haptic feedback avoids all these problems, the vibration alerting the user is almost inaudible, it is direct and natural (haptic feedback is similar to being guided by another human by touch)

Due to the user being blind or partially sighted, light feedback cannot be used. Haptic feedback is the only form of feedback suitable in a built environment or any other environment.

**Although Proxicane can be attached to any limb or even equipment, why has the lower leg been chosen?**

Other ultrasound location devices for the blind have all focused on the device being attached to the arm or hand. Proxicane purpose is to replace the cane so that blind and partially blind users can navigate more discreetly and with a lower cognitive load.

The cane has role of sweeping what is in front of the user in a built environment, the endpoint of the cane which hits the lower part of any environment (such as the floor or lower part of a wall) is what provides a haptic feedback to the user. By attaching Proxicane to the leg, the Proxicane ultrasound sensors would also replace this role.

Logically, majority of the obstacles a user will come across in a built environment would have to be “connected” to the ground. So, sweeping the lower part of the environment using ultrasound would be more efficient in this situation. For example, all chairs and tables must have a leg that touches the ground, however most chairs, tables and other objects do not reach a height that is taller than an average person, so a user could not detect a low table with a cane unless they swept with a cane and detected the legs.

Also, it is more discreet for the user and less to noticeable to others that the device is attached to the lower leg. The upper arms of the user are also free for the user to use in navigation and other activities.

Furthermore, the lower legs can only be used for walking or standing- locomotion activities. Whereas the arms and especially the hands of the user are needed for more finer control and touch. Proxicane being attached to the leg allows the users hands and arms to be free to be used in other activities.

**Proxicane Parts list and technical specifications:**

[include cost of the parts too]

* **Elegoo UNO R3 Board (any Arduino UNO based board can also be used):**

The Arduino project has fully open-sourced both its hardware and software, so similar cheaper boards such as the Elegoo will operate in the same manner as a official Arduino.

The source code of Proxicane will work with the official Arduino Uno and other Arduino based boards, so the user is free to choose any Arduino based board.

<https://www.elegoo.com/product/elegoo-uno-r3-board-atmega328p-atmega16u2-with-usb-cable/> [diagram the board labelled]

Average cost: £7.99 to £14.99

Microcontroller ATmega328

Operating Voltage 5V Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide Pulse Width Modulation output)

Analog Input Pins 6

2 electric current outputs from the 5V and 3.3V pin:

* 5V Electric current: 500MA
* 3.3V Electric current: 50MA

Flash Memory 32 KB of which 0.5 KB used by bootloader

SRAM 2 KB

EEPROM 1 KB

Clock Speed 16 MHz

Working Temperature:  -40°C to 85°C

Net weight: 25g

Dimensions:

* Length:
* Width:
* Height:

The serial interface chip is different to the official Arduino Uno R3 board. The Elegoo board uses an ATMega16U2 instead of the ATMega8U2 chip. Faster transfer rates and more memory.

Programmable using the official Arduino IDE, connect to a computer using a standard USB cable.

[**https://components101.com/sites/default/files/component\_datasheet/Arduino%20Uno%20Datasheet.pdf**](https://components101.com/sites/default/files/component_datasheet/Arduino%20Uno%20Datasheet.pdf)

* **Ultrasonic Distance Sensor (HC-SR04)**

<https://www.piborg.org/sensors-1136/hc-sr04>

[labelled diagram]

Average cost: £3

Input Voltage: 5V

Current Draw: 20mA (Max)

Digital Output: 5V

Digital Output: 0V (Low)

Working Temperature:  -15°C to 70°C

Sensing Angle: 30° Cone

Angle of Effect: 15° Cone

Ultrasonic Frequency: 40kHz

Range: 2cm - 400cm accurate to the nearest 0.3cm

Net weight: 20g

Dimensions:

* Length: 43mm
* Width: 20mm
* Height (with transmitters): 15mm

Will require 4 pins on the Arduino.

* **Flat vibrating Vibration Motor**

<https://www.amazon.co.uk/gp/product/B079121QJ7/ref=ppx_yo_dt_b_asin_title_o07_s00?ie=UTF8&psc=1>

Average cost: £6-7 for 5 pieces

Material: Iron

Working Voltage: 2.5 - 4.0V

Output Speed: 12000RPM

Net Weight: 10g

Dimensions: 10mm x 2.7mm

Maximum start voltage is at an average 2.3v, this is important because without a voltage higher than 2.3v the vibration motor will not start. This is because of the unbalanced weight in the motor, a high voltage is needed to move it. Proxicane needs to be able to deliver power fast, in order for the haptic feedback to be mission critical.

<https://www.precisionmicrodrives.com/vibration-motors/coin-vibration-motors/>

* **Jumper wires**

<https://www.amazon.co.uk/Elegoo-120pcs-Multicolored-Breadboard-arduino-colorful/dp/B01EV70C78/ref=sr_1_3?keywords=jumper+wires&qid=1556015743&s=gateway&sr=8-3>

Average cost: £6 for 120pcs

Only 8 jumper wires are needed to build the Proxicane

Jumper wires are used to rapidly build and prototype. User who want a more permanent solution can use any wire and solder the connections between components.

* **9v battery holder box with on/off switch**

<https://www.amazon.co.uk/Elegoo-120pcs-Multicolored-Breadboard-arduino-colorful/dp/B01EV70C78/ref=sr_1_3?keywords=jumper+wires&qid=1556015743&s=gateway&sr=8-3>

Average cost: £3 each

Dimensions: 6.8 x 3.3 x 2.1cm

Net weight: 10g

* **9v battery**

Can be found in almost any supermarket/ electrical store

The battery must be Alkaline, preferably a long-life battery due to mission critical nature of Proxicane and long durations (4+ hours) that it needs to be on.

These are the core essential parts required for Proxicane to work. The user is free to choose the ideal case for Proxicane. I opted for the cheapest and lightest solution which was to use duct tape. Duct tape provides high strength and waterproof ability. Velcro straps and stickers were used so that Proxicane can be mounted to the lower leg or any equipment.

**Total average cost of a single Proxicane unit (without buying in bulk orders): £18.50**

**Features and Limitations of Proxicane**

[features: can be attached to anything, Velcro straps and Velcro fastenings as long as the vibration motor is in contact with users body]

**Features:**

* Fully open source design and source code. Any user can contribute suggested improvements or bug fixes, after this project is complete, the repository of Proxicane will be made publicly available. <https://github.com/Hussein-Ben/Proxicane>
* A different approach to echolocation for the blind/partially-sighted users: a focus on the device being attached to the lower legs rather than hands or arms
* Focusing on the lower legs, the user is free to use their hand and arms to further enhance navigation or for other activities.
* Allows blind/partially sighted to travel in a built environment more naturally, to react to obstacles passively i.e. without having to use a cane to hit a obstacle.
* An emphasis on discretion, a more discreet device draws less attention to the user, possibly increasing their confidence in navigation
* Only requires a 9v battery, future work on Proxicane will reduce the power consumption
* All components are low-cost and easy to obtain or even salvaged (vibration motors can easily be found in disused smartphones etc..). There are better sensors such as laser range finders, but this was avoided as it was expensive, (above £40 for a single sensor)
* Low cost allows rapid changes to made to the design without fear of financial burden.
* Simple to operate, user only has to turn on device via a switch and attach the device to lower leg, they can start navigating immediately
* Changing the object distance detection can be done by editing the source code, the Arduino only needs to be plugged in
* Accessible source code, only 60 lines of source code which is well commented. With little coding knowledge anyone understand the entire source code of Proxicane
* Haptic feedback, more discrete than a cane (which must by design be loud, as the user hits/sweeps across surfaces), if the user has deafness or reduced hearing haptic feedback would not impair the navigation abilities unlike using a cane
* Ultrasound is not affected by lighting, can work in any lighting condition, unlike a laser range finder or any other sensor that relies on light.
* Haptic feedback provided by the vibration motor is the same feedback used in modern smartphones, Users would most likely familiar with the vibration provided by the motors.
* Low cost, cheaper than a cane. Standard canes on average cost £ 20-24. Even without further optimisations and without buying parts in bulk, Proxicane on average costs £18.50

Examples of current cane prices:

<https://shop.rnib.org.uk/mobility/canes/100cm-folding-guide-cane-pencil-tip.html>

<https://www.amazon.co.uk/White-Aluminum-Mobility-Folding-Sections/dp/B019LOFY5G/ref=sr_1_2?keywords=cane+for+blind&qid=1556033970&s=gateway&sr=8-2>

**Limitations:**

* Different surface materials reflect ultrasound waves differently, this affects the accuracy of readings. For example, fabric materials will absorb some of the ultrasound waves reducing accuracy of object detection (distance to detect an object will be lower). Hard flat surface normally reflects best.
* Due to obstacles having different materials, the detection distance for different obstacles will not be uniform. For example, a sofa will be detected more slower than a closed wooden door, because the sofas material will absorb ultrasound waves more.
* Where the haptic feedback end is attached to the leg, each Proxicane unit has its own haptic feedback end or output. When two Proxicane units are attached to a user’s leg, the haptic feedback ends need to be at least 15cm away from each other, this is to ensure that the user is clear which unit detected a obstacle.
* The problem with multiple haptic feedback outputs on the body is that a user could mistake or misinterpret the wrong reading
* Certain user may be sensitive to haptic feedback
* It is unclear what the effects of long-term haptic feedback is, does the user have reduced sensitivity to it? does it distress the user? [look for papers]
* Proxicane is still a concept, it is not intended to be mission ready yet.
* Not suitable for accurate object Collison detection, designed for day-to-day movement at a walking pace. Should not be used when cycling for example.
* To keep costs low as possible, only cheap ultrasound sensors can be used, it is unclear how long these sensors can endure in real-world situations. More testing is needed.
* Collison detection is not “smart” if the user was talking to a person in front of them, Proxicane would register this as a Collison, there needs to functionality added that allows the user to determine if a collision is a false alarm. But even this functionality would become annoying to the user, the vast number of false collisions the user will come across in a typical day, it is clear that a machine learning model would be needed. A machine learning model can be trained to recognise when collisions are false, thereby reducing the load on the user. In summary, Proxicane needs artificial intelligence
* The current power consumption and battery life of Proxicane does not allow all day collision detection.
* Jumper wires allow rapid changes to be made to the design but are not ideal as a permanent solution as the wires can become loose. Soldering connections together is another feature that users will be encouraged in further iterations of Proxicane.
* Changing the detection distance requires the user to have access to a computer, it currently cannot be changed immediately. Proxicane needs to be completely standalone.
* Climbing up a flight of stairs will cause Proxicane to constantly vibrate as it would assume that there is a Collison in front. The user needs a way to shut down temporarily any false readings. Currently the user can switch off Proxicane manually via a switch on the battery pack.
* In terms of intelligence of the Proxicane device, having real-time intelligent object detection (for example, recognising stairs etc..) would require processing power that not even the high-end tier smartphones of today could provide. When

**Proxicane source code**

Proxicane source code is the extended source code provided by <https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/Ultrasonic_Sensor_HC-SR04.c> this source code was needed as a base to start development from, there is only one standard way of operating a Ultrasound sensor. Total source code memory size is 3kb.

int trigPin = 9; // Trigger - ultrasound waves beamed out

int echoPin = 8; // Echo - listening for the wave that was beamed out

int vibrate\_power = 10; // pin that supplies power to the vibrate function

int collision\_distance = 90; // the distance from an object, the user coul collide with

long duration; // time in microseconds since firing a pulse and recieving a pulse

long distance; // constantly measured - this value is not 100% accurate but it is sufficient for object avoidance

void setup() {

//Serial Port begin

Serial.begin (9600);

//Define inputs and outputs

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(vibrate\_power, OUTPUT);

}

void loop() {

// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

// The sensor is triggered by a HIGH pulse of 10 or more microseconds.

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read the signal from the sensor: a HIGH pulse whose

// duration is the time (in microseconds) from the sending

// of the ping to the reception of its echo off of an object.

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

// Convert the time into a distance

// distance = (time/2) x speed of sound

// The speed of sound is: 343m/s = 0.0343 cm/uS

distance = (duration / 2) \* 0.0343;

// duration is divided by 2 first because the the wave was sent,

// collided with the object, and then returned back to the echo sensor

Serial.print(distance);

Serial.print("cm");

Serial.println();

if (distance < collision\_distance ) { // warn user there going to collide with an object

Serial.println("collision");

digitalWrite(vibrate\_power, HIGH); // physically warn user

delay(100);

} else { // stops false reading

digitalWrite(vibrate\_power, LOW);

}

}

**Testing**

**Proxicane testing criteria:**

Note: each Proxicane device with one haptic feedback end is referred to as a unit

1. Drop test: Can a Proxicane unit function after a drop of 1.5 meters?
2. Detection distance: Can Proxicane detect obstacles (walls, people) within 1 metre?
3. Detection reaction time: how long does it take for Proxicane Unit to provide vibration feedback to the user?
4. False readings: what kind of items and surfaces in the real world that can create a false reading for a Proxicane unit?
5. Impact test: how much force can a Proxicane unit withstand before failure? How much pulling force can the wires endure?
6. Humidity and waterproof, watersplash test: Can a Proxicane unit operate in a humid environment accurately, does humidity and/or water damage affect the accuracy of the ultrasound sensor? A Proxicane unit is normally attached to the leg, will sweat cause any issues?
7. Temperature: highest and lowest temperature a Proxicane unit can operate. Does temperature affect the Ultrasound sensor readings?
8. Does wind affect the ultrasound ability?
9. Do loud sounds in environment interfere with Proxicane?
10. Interference: does other ultrasound sensors and electromagnetic emitters affect the Proxicane ability to detect collisions.
11. UX: general user experience with Proxicane, the difficulty level experienced by the user. Does the user need training to start using Proxicane?
12. Battery life: in the most demanding use how long is the Proxicane battery life expected to last?
13. User movement speed: what is the fastest speed a user can move before the Proxicane unit object detection ability is affected?
14. Object avoidance: are users avoiding collisions with objects? The efficiency of avoiding objects in different types of built environments.
15. Source code: are there any compilation errors or runtime errors? Any unused variables?

**Tests:**

Proxicane has not been tested with blind users yet, but its has been tested with users who were blindfolded to block out vision.

1. I have dropped a Proxicane unit from a height of 2 metres and it is fully operational. Object detection still works. However, after 2 drops the battery covers falls off.

Suggested Improvements: currently as a proof of concept duct tape is used, the battery compartment needs to be less exposed. The user is encouraged to design their own form of casing.

1. The source code variable for detection distance is inaccurate by an average of 100% when applied to real world object detection. For example, if the variable for object detection distance is set at 100cm (so if user is less than 100cm from an obstacle), in real world applications the actual distance a user needs to be before Proxicane alerts them is 50cm. this is due to the different materials ultrasound waves reflect off and the low cost of the Ultrasound sensor, more expensive laser range finders avoid this issue. A user wanting to be alerted to obstacle a metre away would need to set the variable to 200 cm.
2. Reaction time: Proxicane reacts to a Collison in under 1 second. A Collison is defined as a Proxicane unit detecting the distance calculated by the ultrasound sensor as less than 100 cm (or any value defined by the user in the source code). Under 1 second is sufficient at a walking pace in a built environment without moving obstacles such as people. More testing is required to see if the reaction time is sufficient in built environments that have large crowds of people moving around a user.
3. False readings occur only if you place an object in front of sensor momentarily, for example, waving a hand in front of it. The problem is that Proxicane does not consider the thickness of the material, so if a piece of paper or even a leaf is in front of the sensor, Proxicane will detect this as a Collison.
4. 500 grams of weight can be applied to the wires without any connection break. However, reinforcement is needed by having all connections soldered. Currently, Proxicane does not have a case to protect the Arduino. Without a case at the current state of this project, Proxicane should not be used for mission critical object avoidance.
5. Proxicane will not work and be fully damaged if submerged in water. Cannot be used for obstacle avoidance in water or near water (walking besides a beach tide for example). As a proof of concept, I covered a Proxicane unit in duct tape to secure it. Duct tape is waterproof, but because the ultrasound sensor emitters and receivers need to be exposed in order to work, Proxicane cannot be splash proof. Research needs to be done to find a way to make low cost ultrasound sensors waterproof.
6. Speed of sound is temperature-dependent and changes by approximately 0.17% with each degree Celsius according to the Pepperl+Fuchs a company that specialises in electronic sensors(<https://www.pepperl-fuchs.com/great_britain/en/25518.htm>) Proxicane is confirmed to fully operate in temperatures ranging from 10°C to 27°C, this is was temperature possible to test at. The Arduino has a working Temperature between -40°C to 85°C but the ultrasound sensor has a working Temperature between -15°C to 70°C. The required resources needed to test working temperatures to the extremes is not available, however it is safe to assume that Proxicane will no problems working indoors or in average weather conditions outdoors. Temperature must be compensated for, a suggested improvement is that proxicane now needs a temperature sensor as well.
7. Using a hairdryer to simulate wind near the ultrasound sensors has not shown any noticeable differences to the readings, this is the most stress that can be simulated. According to <https://www.pepperl-fuchs.com/great_britain/en/25518.htm> wind speed below 50-61.5 km/h should not affect ultrasound readings.
8. Proxicane has been placed in loud environments of 120 decibels and still operated as usual.
9. Other ultrasound sensors, if in the line of sight of a Proxicane unit will cause severe interference and produce false readings.
10. Proxicane is still a proof of concept, the casing is not available yet. Currently, the only input to the Proxicane unit is on/off switch on the battery pack. The only method that a user can modify the detection distance is via connecting the Proxicane unit to a PC using a USB cable. Short training is required in order to explain what Proxicane is to a new user. Clearly, Proxicane user experience is not intuitive nor is it ready. Many improvements are needed, the Proxicane unit needs to be smaller with a hard case. There needs to be buttons on the case that allow a user intuitively adjust collision detection distance immediately. A user also needs to alert the unit, a temporary sleep button to stop false readings when climbing stairs for example. Priority needs to be given to making a Proxicane unit standalone, requiring no PC for modification to Collison distances. Ideally Proxicane needs intelligence to adapt to users current situation, the processing power and cost required for this is the main setback.
11. Battery life was tested whereby the Proxicane unit was constantly in a state of collision, the haptic feedback end was constantly vibrating. The Proxicane unit lasted for a maximum 3 hours.

**Proxicane development process**

[development process – attempts with web api and the infrared sensor]

**Instructions for building Proxicane:**

**Future Improvements needed for Proxicane**

[reducing size]

[reducing power consumption]

[3d printed case or cut out cardboard for users]

[making it easier for the user to change the object detection distance]

**Before Proxicane – an attempt at an open source refreshable braille display**

[explain current problems with braille displays – outdated tech from 80’s]

[explain your solution – divide and conquer rail approach]

[extend introduction from PPR]

**Braille Reader Specifications**

[extend what is in the PPR]

**Experiments**

[extend Progress to Date from PPR – include justifications]

[explain why I did something every time and why it was abandoned]

**What if braille itself needs to be updated?**

[talks about the pros and cons of alternative system]

[mention how hard braille is]

**Conclusion**

[further work: Arduino uno needs to be replaced with a smaller microcontroller]

[further work: the power requirements need to be lower – i.e double AA or coin cell]