SONAR for the Blind

Tarun Thathvik Smrithi Thudi Vedant Desai

I don’t want the cane…

- **Fixed Length of the cane**

  When open, canes are long and take up a lot of space when used, and make it really uncomfortable sometimes in narrow spaces. Moreover, this rigid structure makes it impossible to reduce the length of the cane as per the users desire.

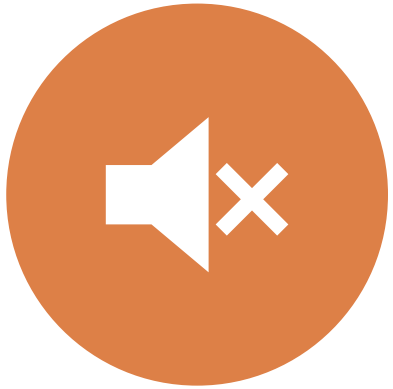
- **Need to hold a cane**

  It is needless to say that using a cane requires one to hold it with a hand, and this restricts them from using this hand fully and normally.

Age old solution

- Bioinspired from bats (sorry, I brought that up :P)
- SONAR (**SO**und **NA**vigation **R**anging): The primitive forms of the modern Sonar was first patented in 1912 by English meteorologist Lewis Fry Richardson.
- Although the actual technology can be used to get a full map of the area scanned through various kinds of echoes, we use it in a more primitive sense, for calculating the distance of the object by checking the time taken by the unique ultrasonic waves to travel from the sensor to the object and back.

How do you feel the distance?



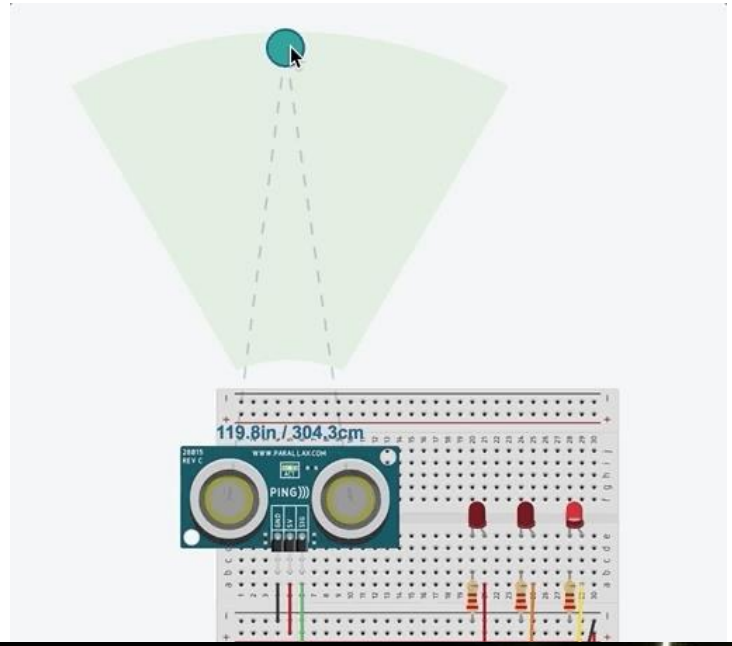
NOT SOUND

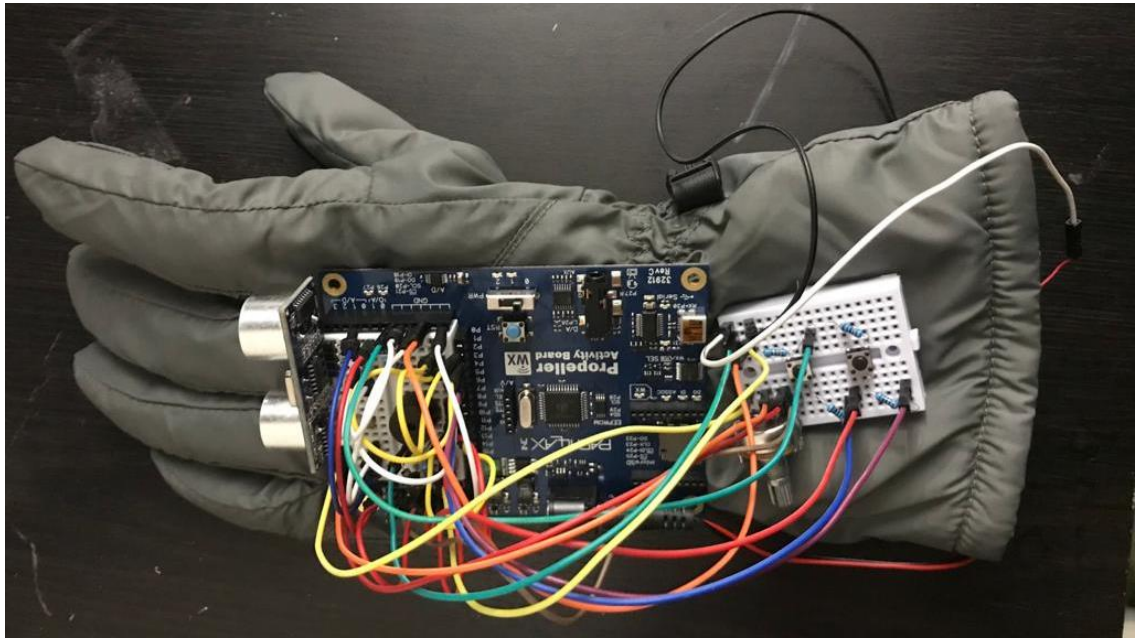
NOT A SIMPLE TOUCH

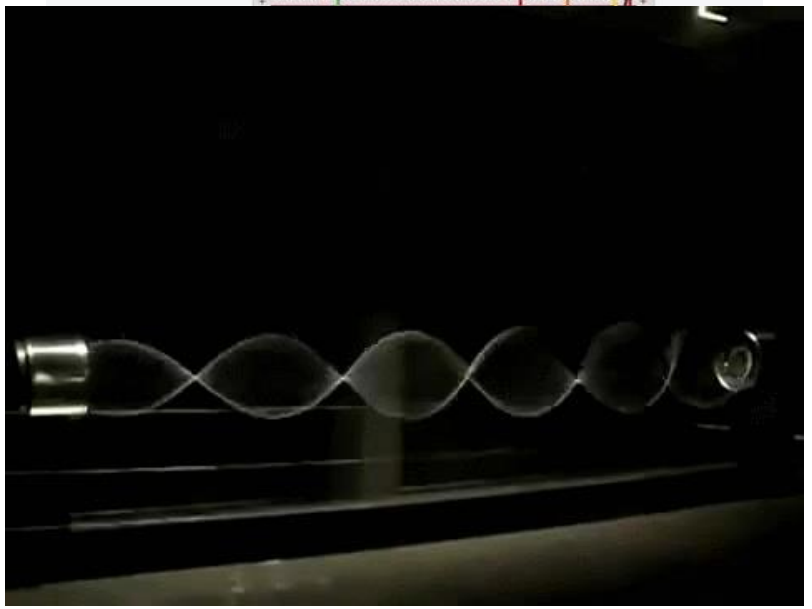
BUT THROUGH VIBRATION.

Putting it together

The Vibration motor vibrates at increasing intensity indicating the closeness of the object from the device/user.



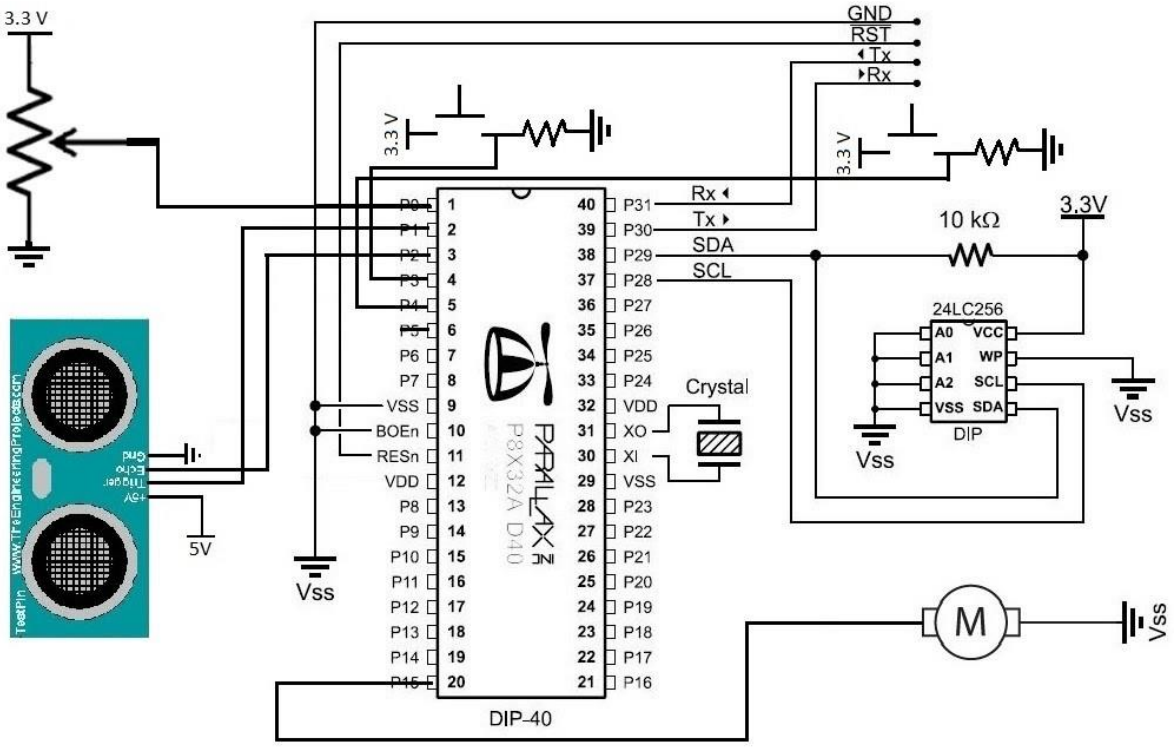




Can I change things?

Two Calibration Modes:

- Distance Calibration Mode
- Vibration Calibration Mode



What we used

#	Components	Cost
1	Parallax Propeller Activity Board	$ 79.00
2	Ultrasonic Sensor [HC-SR04]	$ 4.20
3	Vibration Motors [A00000117]	$ 12.99
4	Piezo Buzzer [A14121600UX0159]	$ 2.19
5	Potentiometer, push button, and other components	$ 10.00
Total		$ 107.28

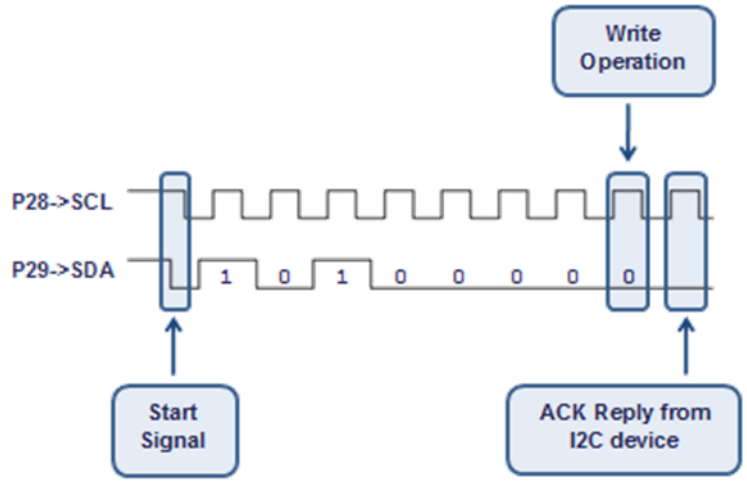
Did we Exploit the Propeller?

- Of the 8 cogs, we utilize 7 Cogs - 4 run continuously while 3 cogs run whenever required.
- EEPROM - I2C protocol (2 wire (SCL,SDA), multi device communication - start condition, I2C address, and read/write bit and ACK from the I2C device followed by data
- ADC (SPI Protocol) - Potentiometer
- DAC - vibration motor intensity control

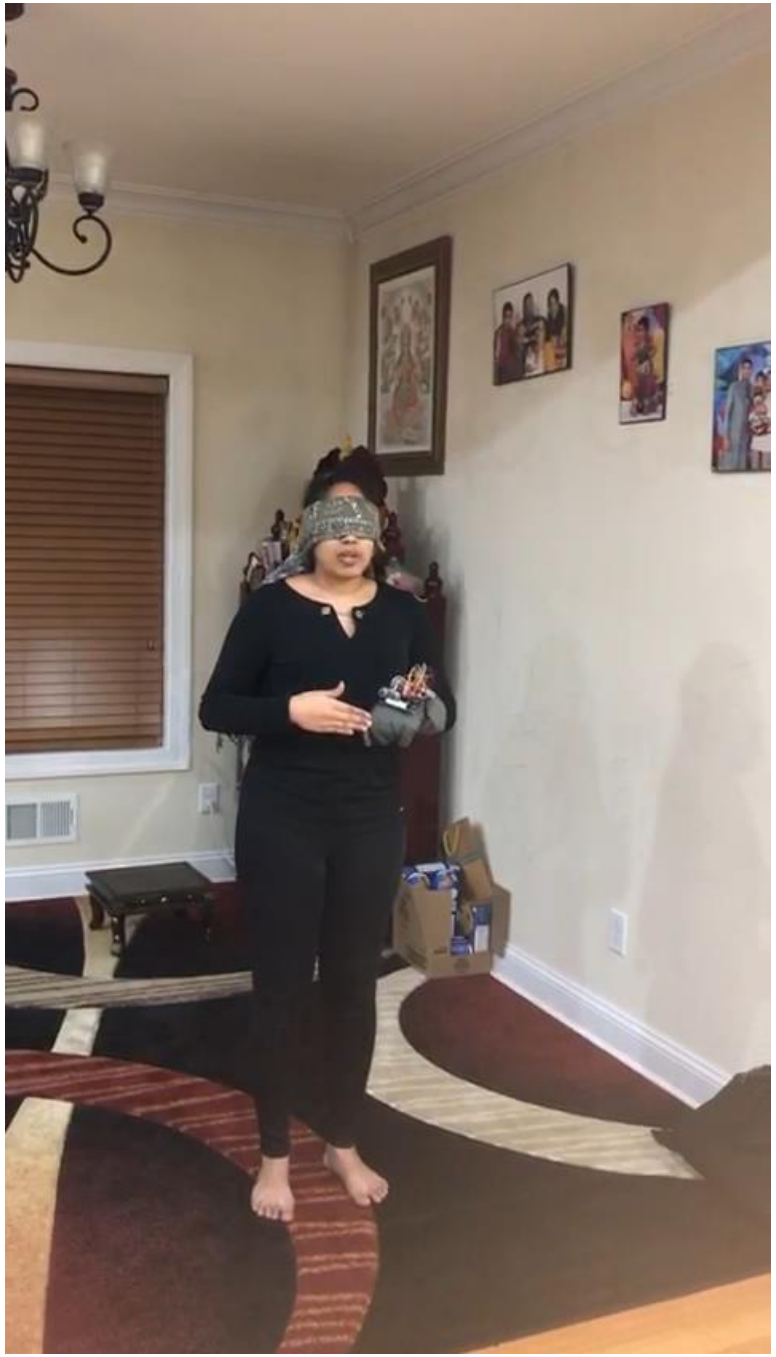
I²C

```
i2c *eeBus;


Int main()                    // Main function
{
    // Set up I2C bus, get bus ID
  eeBus = i2c_newbus(28,   29,    0);
  i2c_in(eeBus,eeAddr,memAddr_firstTime,2,(char*) &check_val,4);
  if (check_val!=1)
  {
    firstTime();
  }


  i2c_in(eeBus, eeAddr, memAddr_max_dist,2,(char*) &max_dist,4);
  i2c_in(eeBus, eeAddr, memAddr_vib_max,2,(char*) &vib_max,4);

}
```

```
void firstTime()
{
  int val =50;
  i2c_out(eeBus,eeAddr,memAddr_max_dist,2,(char*) &val,4);
  while(i2c_busy(eeBus,eeAddr));


  val = 150;
  i2c_out(eeBus,eeAddr,memAddr_vib_max,2,(char*) &val,4);
  while(i2c_busy(eeBus,eeAddr));


  val = 1;
  i2c_out(eeBus,eeAddr,memAddr_firstTime,2,(char*)
&val,4);
  while(i2c_busy(eeBus,eeAddr));
}
```

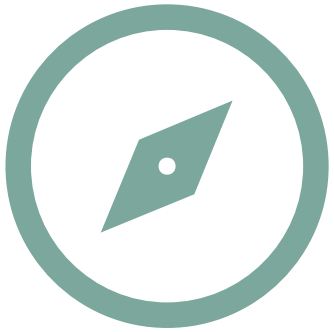


Demo



What next?

Better Vibration Motor



Make it a full-fledged Navigation kit

Thank you!