Forget Me Not
Walking Stick

Final Project Presentation
Robots for Disability

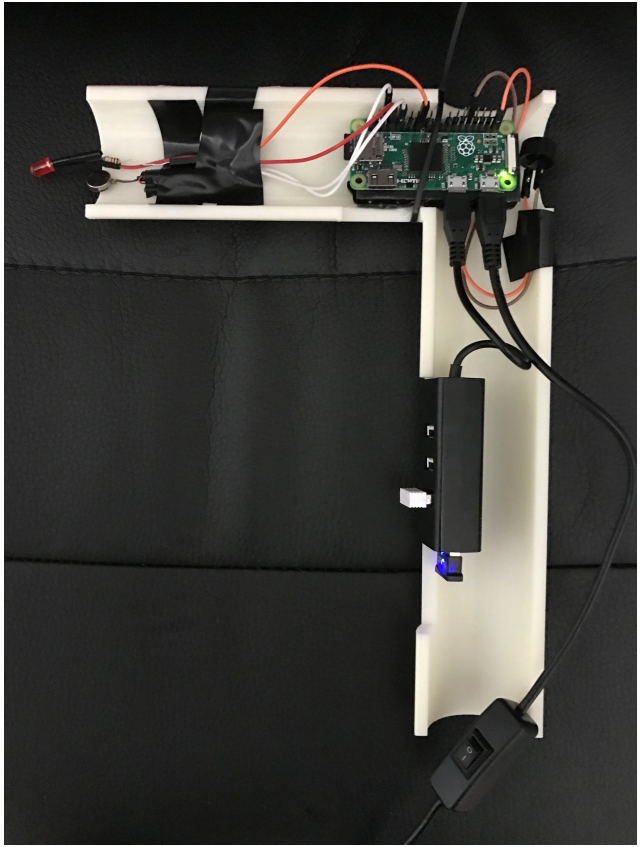
By
Mitravarun Anand
Angad Boralkar
Under the guidance of
Professor Vikram Kapila

Age Related Memory Loss

- Forgetfulness can be a normal part of aging. As people get older, changes occur in all parts of the body, including the brain.
- About 40% of people aged 65 or older have age associated memory impairment—in the United States, about 16 million people.
- Some serious types of memory loss are: Amnestic Mild Cognitive Impairment (MCI),vascular dementia and Alzheimer's disease.
- As a result, people tend to lose their valuables like wallet, keys, phones,etc. Some cases of in-house neglect(eg. In kitchen) also causes serious threat to oneself as well as the well being of others living in the house.

What’s the Forget Me Not Walking Stick?

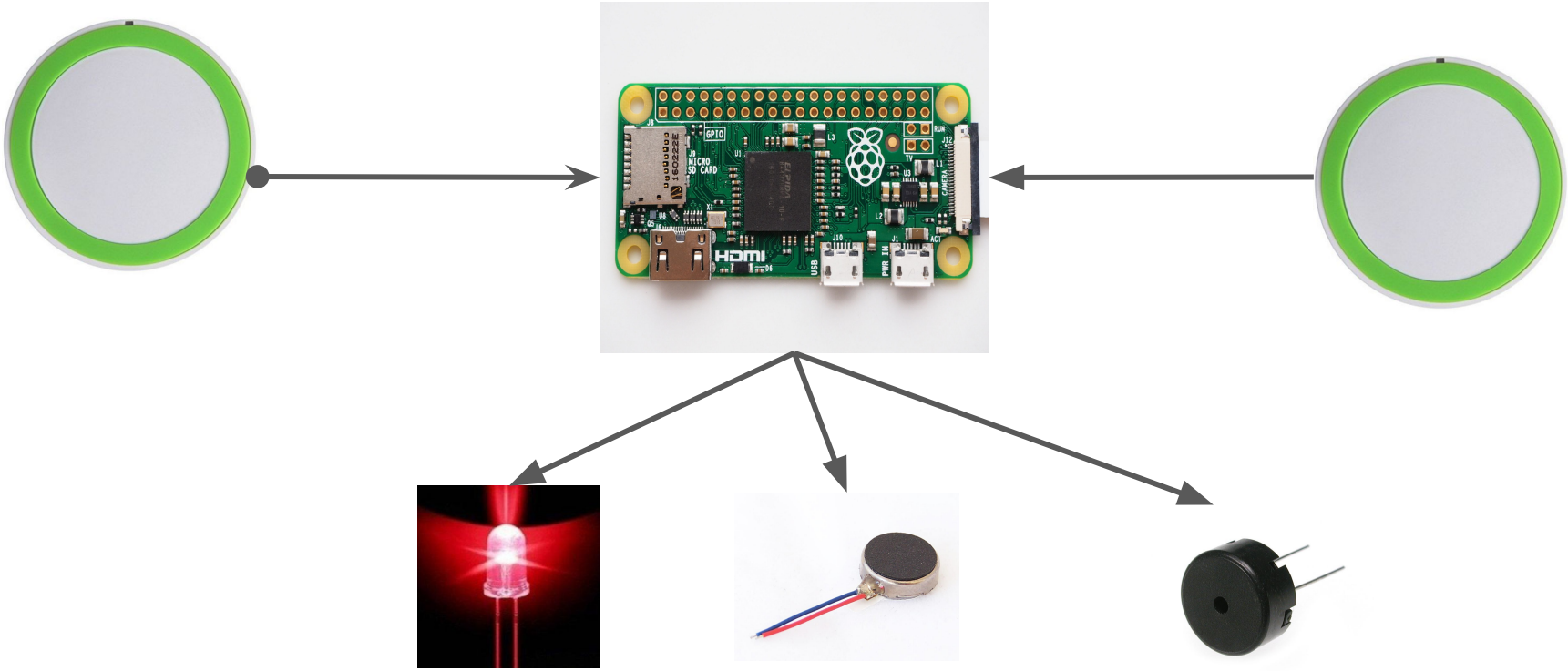
- A stand-alone walking stick that reminds/ warns seniors suffering from any form of memory loss about all the important things in their home.
- Uses Raspberry Pi Zero controller to continuously talk to the iBeacons, that are placed on valuables, or inside an area of the house, like living room or kitchen.
- Can be used to remind if the user forgets valuables (mobile, wallet, keys) and safety aspects in the kitchen.
- When the user comes into proximity of the specific beacon and leaves, the Pi activates warning mechanisms to the user in the form of vibrations, light and sound.



Development

- iBeacon BLE setup with Raspberry Pi Zero
- Distance measurement from RSSI signal of the beacons.
- Haptic feedback to the user with vibration motors.
- Visual feedback with LEDs
- Piezo speaker for those with vision issues.

How it works



Features

Case 1:

- Stick is closest to the object:
  - LED, buzzer, motor are all off. No warning is given.

Case 2:

- Stick goes halfway out of the room:
  - LED blinks 2 times, motor vibrates twice, no buzz.

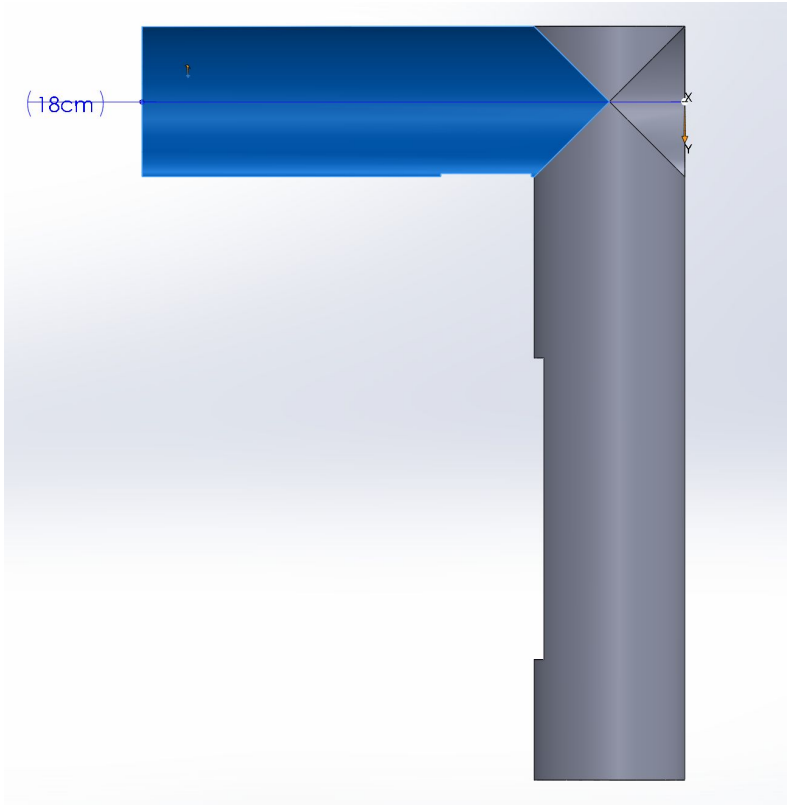
Case 3:

- Stick is almost out of the room:
  - LED blinks thrice, motor vibrates thrice, buzzer gives a short beep.

Case 4:

- Stick is way out of the room:
  - LED blinks once and stays on, motor keeps vibrating, buzzer gives four long buzzes.

Design- Prototype



BLE_Function.py

```
# BLE iBeaconScanner based on https://github.com/adamf/BLE/blo
# JCS 06/07/14

DEBUG = False
# BLE scanner based on https://github.com/adamf/BLE/blob/maste
# BLE scanner, based on https://code.google.com/p/pybluez/sour

# https://github.com/pauloborges/bluez/blob/master/tools/hcito
# https://kernel.googlesource.com/pub/scm/bluetooth/bluez/+/5.
# https://github.com/pauloborges/bluez/blob/master/lib/hci.c#L

# performs a simple device inquiry, and returns a list of ble
# discovered device

# NOTE: Python's struct.pack() will add padding bytes unless y
# should be used for BLE. Always start a struct.pack() format

import os
import sys
import struct
import bluetooth._bluetooth as bluez

LE_META_EVENT = 0x3e
LE_PUBLIC_ADDRESS=0x00
LE_RANDOM_ADDRESS=0x01
LE_SET_SCAN_PARAMETERS_CP_SIZE=7
OGF_LE_CTL=0x08
OCF_LE_SET_SCAN_PARAMETERS=0x000B
OCF_LE_SET_SCAN_ENABLE=0x000C
OCF_LE_CREATE_CONN=0x000D

LE_ROLE_MASTER = 0x00
LE_ROLE_SLAVE = 0x01

# these are actually subevents of LE_META_EVENT
EVT_LE_CONN_COMPLETE=0x01
EVT_LE_ADVERTISING_REPORT=0x02
EVT_LE_CONN_UPDATE_COMPLETE=0x03
EVT_LE_READ_REMOTE_USED_FEATURES_COMPLETE=0x04

# Advertisment event types
ADV_IND=0x00
ADV_DIRECT_IND=0x01
ADV_SCAN_IND=0x02
ADV_NONCONN_IND=0x03
ADV_SCAN_RSP=0x04


def returnnumberpacket(pkt):
    myInteger = 0
    multiple = 256
    for c in pkt:
        myInteger +=  struct.unpack("B",c)[0] * multiple
        multiple = 1
    return myInteger

def returnstringpacket(pkt):
    myString = "";
    for c in pkt:
        myString +=  "%02x" %struct.unpack("B",c)[0]
    return myString

def printpacket(pkt):
    for c in pkt:
        sys.stdout.write("%02x " % struct.unpack("B",c)[0])
```

iBeacon_BLE.py

```
# test BLE Scanning software
# jcs 6/8/2014

import blescan
import sys

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(22,GPIO.OUT)
GPIO.setup(4,GPIO.OUT)
GPIO.setup(26,GPIO.IN)

import bluetooth._bluetooth as bluez

dev_id = 0
try:
    sock = bluez.hci_open_dev(dev_id)
    print "ble thread started"

except:
    print "error accessing bluetooth device..."
        sys.exit(1)

blescan.hci_le_set_scan_parameters(sock)
blescan.hci_enable_le_scan(sock)
try:
        while True:
                returnedList = blescan.parse_events(sock, 5)
                print "----------"
                for beacon in returnedList:
                        if(beacon[0:17] in ["20:91:48:df:0e:99",
                                print (beacon[0:17]+ " " +beacor
                                print beacon
                                a = int(beacon[-2:])
                                #b = 4* a
                                #b = int(4 * a)
                                print a
                                if (a >= 50 and a < 60):
                                        print "Out of Range"
                                        GPIO.output(22,GPIO.HIGH
                                        GPIO.output(4,GPIO.HIGH)
                                        time.sleep(0.5)
                                        GPIO.output(22,GPIO.LOW)
                                        GPIO.output(4,GPIO.LOW)
                                        time.sleep(0.5)
                                elif (a >= 60 and a < 70):
                                        print "Way out of Range"
                                        GPIO.output(22,GPIO.HIGH
                                        GPIO.output(4,GPIO.HIGH)
                                        time.sleep(0.25)
                                        GPIO.output(22,GPIO.LOW)
                                        GPIO.output(4,GPIO.LOW)
                                        time.sleep(0.25)
                                elif (a >= 75):
                                        print "Extremely out of
                                        GPIO.output(22,GPIO.HIGH
                                        GPIO.output(4,GPIO.HIGH)
                                        time.sleep(0.125)
                                        GPIO.output(22,GPIO.LOW)
                                        GPIO.output(4,GPIO.LOW)
                                        time.sleep(0.125)
                                else:
                                        print "Inside range"
except KeyboardInterrupt:
```

button_new.py

```
# test BLE Scanning software
# jcs 6/8/2014

import blescan
import sys

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(22,GPIO.OUT)
GPIO.setup(4,GPIO.OUT)
GPIO.setup(12, GPIO.OUT)

p = GPIO.PWM(12, 100)  # channel=12 frequency=100Hz
p.start(0)

import bluetooth._bluetooth as bluez

dev_id = 0
try:
    sock = bluez.hci_open_dev(dev_id)
    print "ble thread started"

except:
    print "error accessing bluetooth device..."
        sys.exit(1)

blescan.hci_le_set_scan_parameters(sock)
blescan.hci_enable_le_scan(sock)
try:
        while True:
                returnedList = blescan.parse_events(sock, 5)
                print "----------"
                for beacon in returnedList:
                        if(beacon[0:17] in ["20:91:48:df:0e:99","2
                                print (beacon[0:17]+ " " +beacon[-
                                print beacon
                                a = int(beacon[-2:])
                                #b = 4* a
                                #b = int(4 * a)
                                print a
                                if (a >= 50 and a < 55):
                                        print "Out of Range"
                                        for x in range(2):
                                            GPIO.output(22,GPIO.HI
                                            GPIO.output(4,GPIO.HIG
                                            time.sleep(0.25)
                                            GPIO.output(22,GPIO.LO
                                            GPIO.output(4,GPIO.LOW
                                            time.sleep(0.25)
                                elif (a >= 55 and a < 65):
                                        print "Way out of Range"
                                        for y in range(3):
                                            GPIO.output(22,GPIO.HI
                                            GPIO.output(4,GPIO.HIG
                                            time.sleep(0.25)
                                            GPIO.output(22,GPIO.LO
                                            GPIO.output(4,GPIO.LOW
                                            time.sleep(0.25)

                                        for dc in range(0, 2, 1):
                                                p.ChangeDutyCycle(
                                                time.sleep(0.1)
                                        for dc in range(2, -1, -1)
                                                p.ChangeDutyCycle(
```

Cost Analysis

Component	Cost ($)	Cost for Mass Production ($)
Raspberry Pi Zero	4.99	2.99
Beacons (x2)	23.99	15.99
3D printing	10.00	3.00
Bluetooth 4.0 Dongle	5.99	3.99
Power Source	12.00	6.00
Vibration motor	4.99	3.99
Miscellaneous	5.00	2.00
Total	**66.96**	**36.96**

Advantages

- Extremely easy to use with a very small learning curve.
- No need to have a smartphone.
- Plug and Play operation for use with additional beacons.
- Can be used by a person who is blind.
- Low cost microcontroller and components.
- BLE is highly energy efficient, beacons can go years before replacement.
- Dedicated on/off switch.
- Additional USB ports: charge other electronics.

Disadvantages

- Cheaper iBeacons drops broadcasting signals with too many obstacles.
- Highly accurate beacons are available, but are expensive.
- No audio output

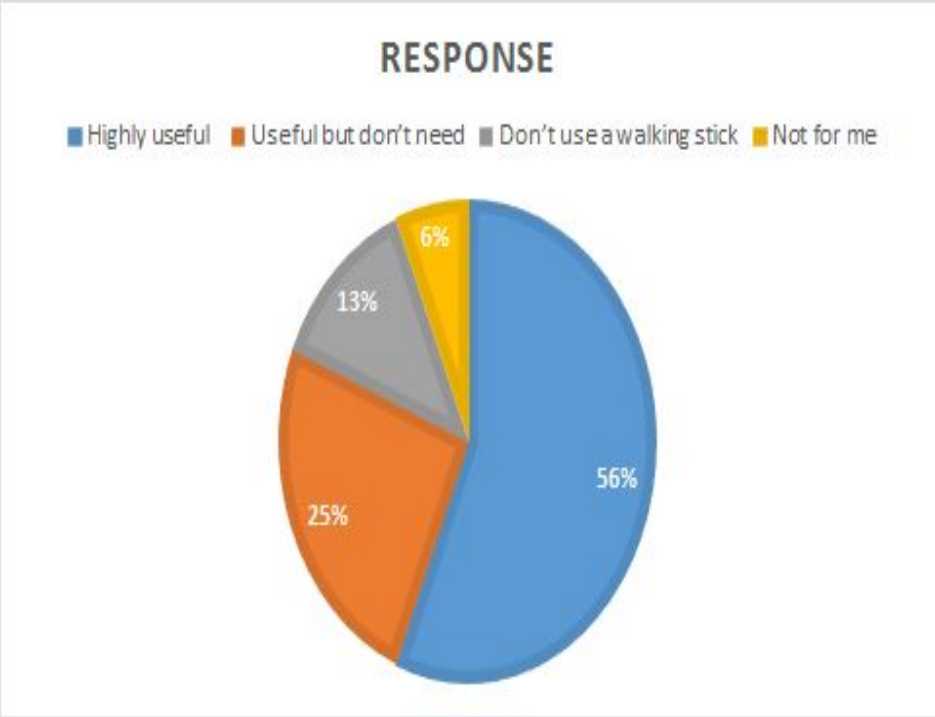
Future Work

- Use better quality beacons.
- Interfacing with home automation systems.
- Interfacing with devices like Amazon Echo to remotely give voice outputs that could assist the users to be mindful of their surroundings.

Validation

Comments:

- Many related to the problem and felt the need for it.
- People are ready to pay upto $100.
- Different color.
- Not on a walking stick.

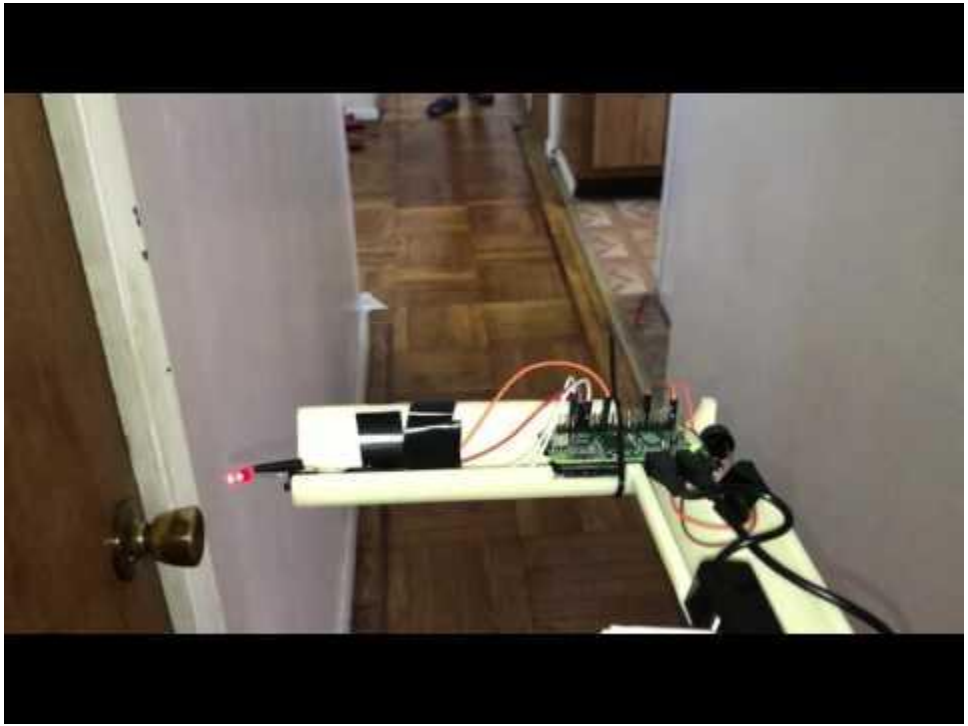


Improvements suggested:

- Smaller form factor.
- Speaker instead of buzzer.

Stats based on talking to 16 people aged above 60, at Bayridge Senior Center

Demo




Questions?