**RASPBERRY PI BASED SMART WATCH**

by

NAMAN ARORA 16BLC1030

ARPAN SATPATHI 16BLC1124

KARANVEER RAGHUVANSHI 16BLC1128

A project report submitted to

**Dr. VETRIVELAN. P**

**SCHOOL OF ELECTRONICS ENGINEERING**

in partial fulfilment of the requirements for the course of

**CSE4017 – SOFTWARE HARDWARE CO-DESIGN**

in

**B.Tech ELECTRONICS AND COMPUTER ENGINEERING**

****

**VIT CHENNAI**

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**APRIL 2019**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “**RASPBERRY PI BASED SMART-WATCH”** is a bonafide work of **NAMAN ARORA(16BLC1030),ARPAN SATPATHI(16BLC1124)** and **KARANVEER RAGHUVANSHI(16BLC1128)** who carried out the Project work under my supervision and guidance.

**Dr. VETRIVELAN.P**

Associate Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

**ABSTRACT**

A smart watch is a unique device that helps the user wearing it to monitor various notifications of his android phone through this watch and alter them. Therefore the situational analysis is more focused to smartphone as the baseline comparison, rather than desktop PC, laptop or tablet. Given the phone has been picked out from pocket and now is on hands, a person still needs to do some steps such as unlocking the phone, finding the app. The number of steps involved are high and thus a smart watch device is helpful. Smart Watch that are already in this market are advanced and costly. This project proposes a smart watch which not only shows time but provides additional features like health monitoring, notification display at a much cheaper price. This watch is programmed using raspberry pi's open source libraries and different APIs and functions. This watch is embedded in various sensors that track and interpret user's actions into information. It provides the user another way to interact with an android smartphone. This watch can also be personalized according to the user's requirement for the functionality.

**ACKNOWLEDGEMENT**

We are extremely grateful to **Dr. Sivasubramanian. A,** Dean of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Programme Chair **Dr. Susan Elias** for his support throughout the course of this project.

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Vetrivelan.P,** Associate Professor, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**NAMAN ARPAN KARANVEER**

**INTRODUCTION**

Though smart watch has been commercially available since early 80’s, it has not gained much publicity or interest from public consumer. However in the past one year, it has gained significant momentum and 2013 was even said by analysts “could have been year of-the Smart Watch”. That momentum was signified by the release of several smart watch products such as Pebble, Razer Inc, Archos, LG, Motorola and even Google Android Wear. This paper is trying to explore what smart watch can or should do in order to save people time, by making some daily tasks processing easier and more efficient. One focus area is how the “human-smart watch” interaction should be designed in order to achieve that time-saving objective. The design is based on what technology is available today as well as what has been patented.

**OBJECTIVES AND GOALS**

* Design a raspberry pi based smart watch.
* The watch should be able to read all the notifications that the user receives on his/her mobile device.
* The watch should be able to download files on a remote device such as a laptop by giving commands to it.
* The watch should be able to keep a track on the number of steps walked by the wearer.

**BENEFITS**

We live in an age where technology is prominent.  With the swipe of a finger, mobile phone owners are able to access an incredible amount of information within seconds.  They can check their email, upload photos immediately after taking them, purchase an item online, and transfer funds to their bank account with very little effort.

Today’s smart phones send and receive information rapidly.  Instant accessibility to data has revolutionized the way people live their day-to-day lives and raised the bar where customer service is concerned.  Rather than wait for an agent to assist over the phone, consumers can get the help that they need through live chat options, social media, and email.

So, if a smart phone can do all that alone, why would you need a smartwatch?

Smartwatches were designed to enhance and improve smart phone usage.  The two work hand in hand or hand and wrist, rather.  Bluetooth technology allows the devices to work simultaneously.

As new smartwatch models emerge on the marketplace, smart phone owners are given more and more options to choose from.  Here are some of the many advantages of having a computer on your wrist:

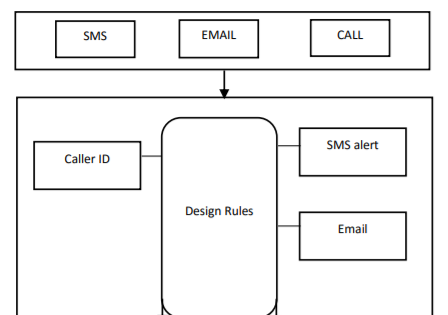
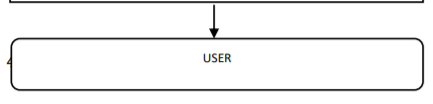
* **Convenience.**Let’s face it.  People like to be well-informed.  They want to feel connected.  That’s what makes the smartwatch so appealing.  Imagine being able to do most of the things you do on a smart phone without having to pull out your phone to do them.  Smartwatches allow you to make and receive calls and receive updates.  You can receive text messages, alerts, and updates without being extremely obvious about it.
* **Affordability.**  Unlike some new technologies, smartwatches are rather affordable.  They complement smart phones so more and more cell phone companies are selling and promoting them.  Even the most highly anticipated models are extremely affordable.  As more and more companies jump onto the smartwatch bandwagon, functionality will improve and prices will drop significantly.
* **Functionality.**Some smartwatch models have built-in flash drives.  Others have voice and gesture control, Cloud data, and built-in activity trackers.  Many smartwatches allow you to text and tweet from your wrist.  Imagine being on the treadmill at the gym.  You no longer have to haphazardly pull your phone out to see who is calling or texting you.  One look at your smartwatch can tell you whether you need to answer the phone or wait until you’re done working out.
* **Discretion.**  Everyone wants to be connected but let’s face it, there are some places where phones are inappropriate.  Despite wanting to be available to your family or clients, it just isn’t possible in some places.  A smartwatch makes it easier to check messages on the fly.
* **Custom Firmware.**App developers have free reign when it comes to designing for the smartwatch.  Companies encourage innovation by providing guides detailed how to “hack” the watches software.  Much like smart phones, the app market for smartwatches could become larger than life.

**FEATURES**

* The watch is able to read all the notifications that the user receives on his/her mobile device.
* The watch is able to download files on a remote device such as a laptop by giving commands to it.
* The watch is able to keep a track on the number of steps walked by the wearer.

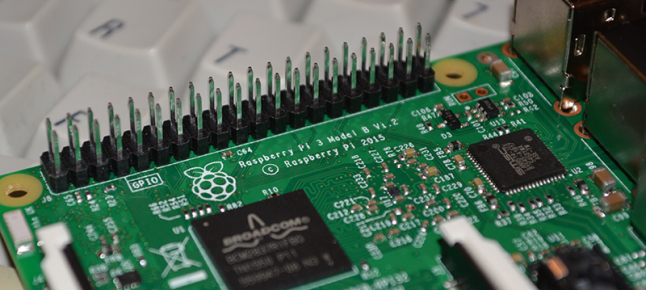
**SYSTEM DESIGN**

**2.1 BLOCK DIAGRAM**

** **

**HARDWARE:**

**Raspberry pi 3 :**



**Raspberry Pi 3 Specifications**

**SoC:** Broadcom BCM2837  
**CPU:** 4× ARM Cortex-A53, 1.2GHz  
**GPU:** Broadcom VideoCore IV  
**RAM:** 1GB LPDDR2 (900 MHz)  
**Networking:** 10/100 Ethernet, 2.4GHz 802.11n wireless  
**Bluetooth:** Bluetooth 4.1 Classic, Bluetooth Low Energy  
**Storage:** microSD  
**GPIO:** 40-pin header, populated  
**Ports:** HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

**Mobile Phone:**

Brand: Xaomi

Model: Mi A1

**SOFTWARE IMPLEMENTATION**

This project uses a number of open source projects and apps to work properly:

* [Pushbullet App](https://play.google.com/store/apps/details?id=com.pushbullet.android&referrer=utm_source%3Dpushbullet.com) - Send your mesages and files across different paltforms using a single app!
* [Pushbullet API](https://docs.pushbullet.com/) - API used for enabling cross platform communication.
* [PhonePI App](https://play.google.com/store/apps/details?id=com.phonepi&hl=en_IE) - For streaming sensor data to our Raspberry Pi.
* [PhonePI Sample Server](https://github.com/priyankark/PhonePi_SampleServer) - Sample server from the creator of the awesome app.
* [Python 3.6.8](https://www.python.org/downloads/release/python-368/) - Any other version of Python 3 should aslo work

**Source code**

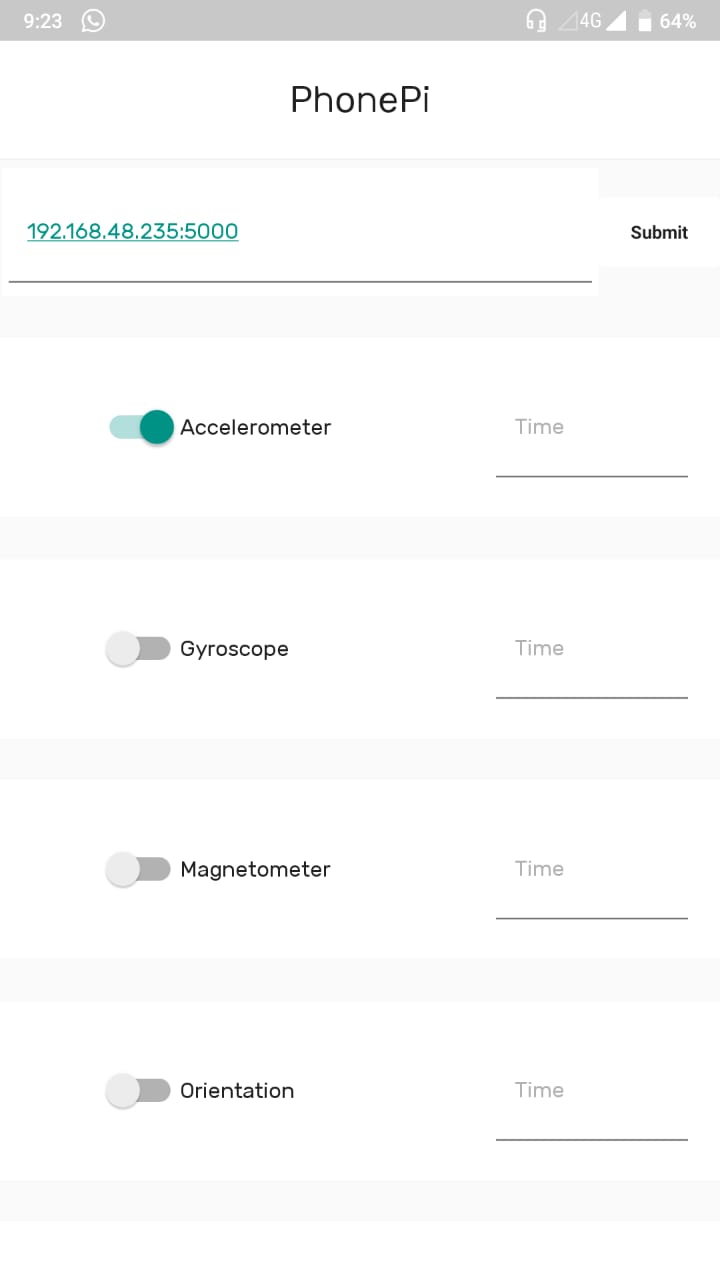
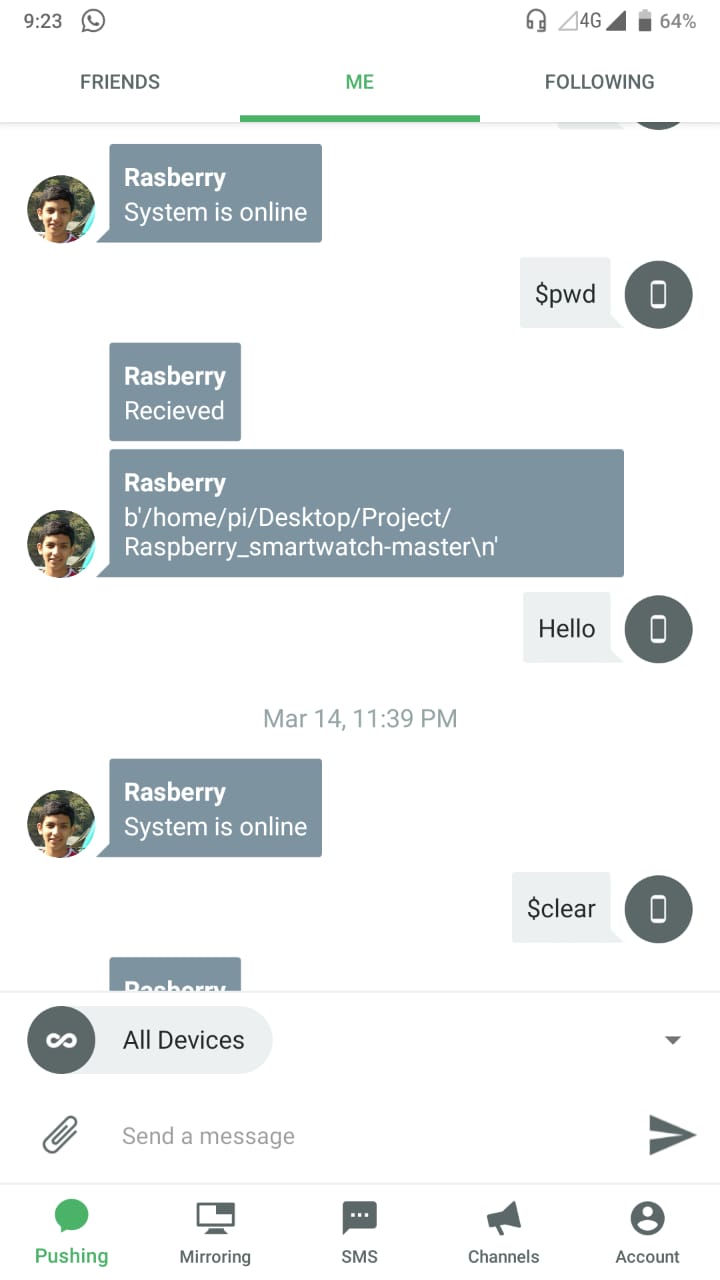
- To get all the notifications on the smart watch:

|  |
| --- |
| import asyncio |
|  | import os |
|  | import sys |
|  | import traceback |
|  | import subprocess |
|  |  |
|  | sys.path.append("..") # Since examples are buried one level into source tree |
|  | from asyncpushbullet import AsyncPushbullet, InvalidKeyError, PushbulletError, LiveStreamListener |
|  |  |
|  |  |
|  | async def \_run(): |
|  | #MY API KEY |
|  | api\_key = "#YOUR API KEY" |
|  | async with AsyncPushbullet(api\_key) as pb: |
|  | async with LiveStreamListener(pb) as pl: |
|  | print("Awaiting pushes...") |
|  | push = await pb.async\_push\_note(title="Rasberry", body="System is online") |
|  | print("Initialization response sent") |
|  | async for data in pl: |
|  | if('file\_url' in data): |
|  | #For text to-do list feature in RPi |
|  | print('\n') |
|  | cmd = 'curl '+data['file\_url'] |
|  | name = data['file\_name'] |
|  | print("Filename = ",name) |
|  | os.system(cmd) |
|  | print('\n') |
|  | elif('title' in data and data['title']=="Rasberry"): |
|  | #For response message by rasberry |
|  | pass |
|  | elif('source\_device\_iden:nickname' in data and 'body' in data): |
|  | #For data sent via Pushbullet |
|  | if('title' in data and data['title']=="Rasberry"): |
|  | pass |
|  | else: |
|  | print('\n') |
|  | print(data['source\_device\_iden:nickname']) |
|  | print(data['body']) |
|  | push = await pb.async\_push\_note(title="Rasberry", body="Recieved") |
|  | print('\n') |
|  | if((data['body'])[0]=="$"): |
|  | print("Output:") |
|  | print('\n') |
|  | os.system((data['body'])[1:]) |
|  | cmd=(data['body'])[1:].split() |
|  | push = await pb.async\_push\_note(title="Rasberry", body=subprocess.Popen( cmd, stdout=subprocess.PIPE ).communicate()[0]) |
|  | elif(data['body']=='exit'): |
|  | sys.exit() |
|  | print('\n') |
|  | elif('push' in data and 'title' in data['push'] and 'body' in data['push']): |
|  | #For data sent via whatsapp |
|  | print('\n') |
|  | print(data['push']['title']) |
|  | print(data['push']['body']) |
|  | print('\n') |
|  | else: |
|  | print("Got a push:", data) |
|  |  |
|  | loop = asyncio.get\_event\_loop() |
|  | loop.run\_until\_complete(\_run()) |

**To count the number of steps:**

|  |
| --- |
| from flask import Flask  import matplotlib.pyplot as plt |
|  | from flask\_sockets import Sockets |
|  |  |
|  |  |
|  | app = Flask(\_\_name\_\_) |
|  | sockets = Sockets(app) |
|  |  |
|  | @sockets.route('/accelerometer') |
|  | def echo\_socket(ws): |
|  | f=open("accelerometer.txt","w") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  |  |
|  | @sockets.route('/gyroscope') |
|  | def echo\_socket(ws): |
|  | f=open("gyroscope.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/magnetometer') |
|  | def echo\_socket(ws): |
|  | f=open("magnetometer.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/orientation') |
|  | def echo\_socket(ws): |
|  | f=open("orientation.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/stepcounter') |
|  | def echo\_socket(ws): |
|  | f=open("stepcounter.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/thermometer') |
|  | def echo\_socket(ws): |
|  | f=open("thermometer.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/lightsensor') |
|  | def echo\_socket(ws): |
|  | f=open("lightsensor.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/proximity') |
|  | def echo\_socket(ws): |
|  | f=open("proximity.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  | @sockets.route('/geolocation') |
|  | def echo\_socket(ws): |
|  | f=open("geolocation.txt","a") |
|  | while True: |
|  | message = ws.receive() |
|  | print(message) |
|  | ws.send(message) |
|  | print(message, file=f) |
|  | f.close() |
|  |  |
|  |  |
|  |  |
|  | @app.route('/') |
|  | def hello(): |
|  | return 'Hello World!' |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | from gevent import pywsgi |
|  | from geventwebsocket.handler import WebSocketHandler |
|  | server = pywsgi.WSGIServer(('0.0.0.0', 5000), app, handler\_class=WebSocketHandler) |
|  | server.serve\_forever()   |  | | --- | |  | |  | from scipy.misc import electrocardiogram | |  | from scipy.signal import find\_peaks | |  | import numpy as np | |  |  | |  | #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Writing to File\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |  | with open("accelerometer.txt", "r") as filestream: | |  | with open("answers.txt", "w") as filestreamtwo: | |  | for line in filestream: | |  | currentline = line.split(",") | |  | total = str((float(currentline[0])\*\*2 + float(currentline[1])\*\*2 + float(currentline [2])\*\*2)\*\*-2) + "\n" | |  | filestreamtwo.write(total) | |  |  | |  | #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |  |  | |  |  | |  | a = [] | |  | b = [] | |  | with open("answers.txt", "r") as m: | |  | for index,line in enumerate(m): | |  | a.append(index) | |  | b.append(line) | |  |  | |  |  | |  | b = list(map(lambda s: s.strip(), b)) | |  | map(float, b) | |  | b = [float(i) for i in b] | |  | b = [i\*100000 for i in b] | |  | b = [int(float(i)) for i in b] | |  |  | |  |  | |  | x = electrocardiogram()[b] | |  | peaks, \_ = find\_peaks(x, height=-0.17) | |  | print(len(peaks)) | |  | plt.plot(x) | |  | plt.plot(peaks, x[peaks], "x") | |  | plt.plot(np.zeros\_like(x), "--", color="gray") | |  | plt.show() | |

**APP SCREENSHOTS:**

**4. CONCLUSION AND FUTURE WORK**

**4.1 CONCLUSION**

* The Raspberry pi based smart watch was built and implemented.
* The system is targeted at students and office employees who are always on the run and are very busy.
* The prototype developed can display phone notifications and can even act as a step counter.
* The preliminary test results are promising.

**4.2 FUTURE WORK**

* More number of sensors can be incorporated and OLED display can be attached as well.
* Voice operated system can be implemented
* Adding confirmation commands to the voice recognition system.
* Adding an anti-theft feature which can detect if the phone is being taken without permission.

**5. REFERENCES**

1. www.google.co.in
2. [www.circuitstoday.com](http://www.circuitstoday.com)
3. [www.fadooengineers.com](http://www.fadooengineers.com)
4. [www.electroschematics.com](http://www.electroschematics.com)
5. [www.engineersgarage.com](http://www.engineersgarage.com)
6. [www.wikipedia.com](http://www.wikipedia.com)
7. [www.electrosome.com](http://www.electrosome.com)