**Technology Report**

**For**

**DigitalDashboard**

**Version 1.0 in progress**

**Prepared by:**

**Karan Raj Kanwar**

**Zhill Patel**

**Darren Prong**

**Humber College**

**School of Applied Technology**

**February 6, 2019**

# Declaration of Group Authorship

We, Karan Raj Kanwar, Zhill Patel and Darren Prong confirm that this work submitted for assessment is our own and is expressed in our own words. And uses made within it of the works of any author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of references used is included. Please refer to chapter 2 Overview for the work breakdown.

**Signed:**

**Student Number:**

**Signed:**

**Student Number:**

**Signed:**

**Student Number:**

**Date:**

# Abstract

In today’s society people enjoy knowing statistics for their travelling periods, but having a bike or a scooter can’t provide real-time information or entertainment. A store-bought scooter or bike doesn't come with a tracker or entertainment but with the DigitalDashboard you can track your speed and listen to music at the same time. Just by simply mounting the portable device on the handle or the body of the source of transportation you can start receiving the real-time travelling speed, location, and music for entertainment purposes. We will demonstrate how our device gets mounted on a bike and scooter, how to use the DigitalDashboard device and its mobile application counterpart.

Table of Contents

[Declaration of Group Authorship 2](#_Toc2872417)

[Abstract 3](#_Toc2872418)

[Revision History 5](#_Toc2872419)

[1.Introduction 7](#_Toc2872420)

[2. Overview 9](#_Toc2872421)

[2.0.1 Database and work breakdown 9](#_Toc2872422)

[2.0.2 Application and work breakdown 9](#_Toc2872423)

[2.0.3 Hardware and work breakdown 10](#_Toc2872424)

[2.0.4 Database and work breakdown revised (2019-03-06) 10](#_Toc2872425)

[2.0.5 Application and work breakdown (2019-03-06) 11](#_Toc2872426)

[2.0.6 Web/Hardware and work breakdown (2019-03-06) 12](#_Toc2872427)

[3. Build Instructions 13](#_Toc2872428)

[3.0.1 System Diagram 13](#_Toc2872429)

[3.0.2 Bill of Materials/Budget 14](#_Toc2872430)

[3.0.3 Time Commitment 16](#_Toc2872431)

[3.0.4 Mechanical Assembly 17](#_Toc2872432)

[3.0.5 PCB/Soldering 18](#_Toc2872433)

[3.0.6 Power Up / Raspberry Pi setup 19](#_Toc2872434)

[3.0.7 Case 20](#_Toc2872435)

[3.0.8 Unit Testing 21](#_Toc2872436)

[4. Results 22](#_Toc2872437)

[5. Data 23](#_Toc2872438)

[6. Analysis 25](#_Toc2872439)

[7. Conclusion 27](#_Toc2872440)

[8. Bibliography 29](#_Toc2872441)

[9. Appendix A: Glossary 31](#_Toc2872442)

[10. Appendix B: Analysis Models 33](#_Toc2872443)

[11. Appendix C: Calculations 35](#_Toc2872444)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Karan, Zhill & Darren | 2019-02-10 | Added declaration of authorship, abstract and introduction | 0.1 |
| Karan, Zhill & Darren | 2019-03-07 | Added the work breakdown | 0.1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1.Introduction

The industry related to our device is the health, fitness and personal transport. The reason why it is considered for health and fitness is because of the adaptability of the device and its small portable form factor. As a personal transportation device it can be easily mounted/installed onto a cylindrical surface. This report includes the creation of the DigitalDashboard, device creation procedure, software used and gathering/analyzing results.

The solution we came up with is a portable mounting digital dashboard, this is fairly small to carry and mounts onto almost all circular surfaces some being the handle of a bike or scooter. This device can tell you your speed, location, altitude, longitude and can play music/radio all while being mounted on a surface.

The main problem is that children’s bikes and scooters don’t come with a digital screen for tracking or any source of entertainment. This we solve by allowing users to mount the DigitalDashboard onto their scooter or bike for tracking and or playing music, this can be use on consumer or professional basis.

# 2. Overview

### 2.0.1 Database and work breakdown (2019-01-24)

For our database we will be using the Firebase Real-time database. After some research it seems the Raspberry Pi and the mobile phone can both utilize Firebase and can be connected to move data around. Karan will be handling setting up the required entities from both mobile and the Raspberry Pi device to the database. Zhill will be making sure that the communication of data between the Raspberry Pi and the mobile application is working and that the data is stored in the database. Darren will be creating and modifying the FM radio (online/offline) functionality and hardware connectivity (TEA5767+amplifier connecting to our database and respond to CRUD operations accordingly). As well, a built in MP3 player; all to be used in conjunction with the LCD application.

### 2.0.2 Application and work breakdown (2019-01-24)

For our mobile application we will be working with Android Studio version 3.3. There will be 3 major components of creating the app. First one being creating a clean layout for the UI and having an aesthetically pleasing icon. The second component being the logic working correctly and having a mobile device communicating with the Raspberry Pi. Lastly getting the mobile device to push and pull data from or to the database. Karan will be handling database creation and the implementation in the application source code. Zhill will be handling some of the functionality of the application, for example, making sure the application will show the data that it got from the sensors. Darren will be handling loading online radio streams when available, and receiving, displaying and saving all local radio stations on the FM band; as well, loading potential MP3’s for the users’ device directly to the application. Our main system feature would be grabbing the speed from the accelerometer and displaying it on the screen, and having the functionality to play music all in device. For the application to run at its best performance an Android device running on 7.0.0 is required.

### 2.0.3 Hardware and work breakdown (2019-01-24)

For setting up our hardware we are going to need a practical sized enclosure for our devices which still holds its purpose of being a portable screen, also we need to attach everything together to get it all to function correctly. We are also required to create the GUI and program on the Raspberry Pi to get our application to work. Karan will be working on the enclosure and the program. Zhill will be working on the hardware portion of the sensors, making sure that it will be able fit inside the enclosure properly. Darren will testing the prototype at each stage of development and also assisting Zhill in fabricating a new PCB with all three sensors and necessary accessories (antenna, AUX port, speakers, etc.) integrated together into a single component. For the application to run at peak performance the latest raspberry pi 3b is required.

### 2.0.4 Database and work breakdown revised (2019-03-06)

For our database we are utilizing Google Realtime Firebase database for both the mobile application and the Raspberry Pi B database. The mobile application allows for reading and writing into the database, our plan was to implement a music database data structure but we ended up discarding that idea as it isn’t really need. We have a user registration and sensor data structure which saves the UID and allows users to access their own information and sensor readings, the sensor reading just stores the UID and the sensor information and can be pulled up when required by user. As for the Raspberry pi database we are in the works on creating database for sensor and user, the code will be written in python as it is easier to create the database with python. Karan has completed the app and has got it working also communicating with the firebase database. Zhill will be working with the new amplifier design in Fritzing; integrating all three PCB designs and sensors unto the final PCB. The components will be supplied by Darren, and tested by both team members. Next step for Darren is to finalize the design with Zhill and fabricate a new PCB with all three sensors and necessary accessories (antenna, AUX port, speakers, etc.), then test the connection with our Firebase Database via our application on the Raspberry Pi. The final enclosure is still to be designed, as this needs the final PCB dimensions in order to be accurate.

### 2.0.5 Application and work breakdown (2019-03-06)

Our device has several pieces which can be separated into small work projects for each team member. Karan has completed the majority of the applications functionality, however, our main display GUI (the one is mainly used on the LCD) is still in development. Zhill will be working on the hardware portion, integrating Darren’s design into the final PCB, as well, design of the mounting points will have to be taken into consideration. Darren has to complete the database design, so that when functional, data is received, perceived and acted upon by the application (i.e. tuning the FM radio sensor). As well, Darren has to assist Zhill in designing and testing the final PCB designs and making sure they conform to the enclosure.

### 2.0.6 Web/Hardware and work breakdown (2019-03-06)

For our hardware at this point of the semester we have full connectivity between the mobile application and the Real-time Firebase database but the Raspberry pi still needs a case which can hold the pi, global positioning sensor, accelerometer sensor, speaker and the custom built PCB. So far Karan has started on the GUI creation for the raspberry pi and is trying to get it to communicate to the Firebase database. Both Zhill and Darren have assisted Karan in the applications development into a functional software platform for our device.

# 3. Build Instructions

# 3.0.1 System Diagram

# [System Diagram](https://raw.githubusercontent.com/KaranRajKanwar/DigitalDashboard/master/Blog%20Images/SystemDiagram.png)

# 3.0.2 Bill of Materials/Budget

| **Part Name** | **Part Number** | **Source** | **Price with shipping & tax** |
| --- | --- | --- | --- |
| Raspberry Pi 3 Model B | B01CD5VC92 | [Amazon](https://tinyurl.com/yd8zt33y) | $55.99 |
| Raspberry Pi Case | B01D8368RY | [Amazon](https://tinyurl.com/y7lkpouc) | $13.99 |
| Raspberry Pi 5V Power Supply | B06Y431Y27 | [Amazon](https://tinyurl.com/ya4kcvr3) | $19.13 |
| SanDisk 32gb microSDXC card | B073JWXGNT | [Amazon](https://tinyurl.com/ydhm22or) | $19.65 |
| 2 Mini Raspberry Pi Heatsinks | B014KKY19G | [Amazon](https://tinyurl.com/yawe5tvz) | $11.98 |
| HDMI Cable | B014I8SSD0 | [Amazon](https://tinyurl.com/y6ups5wd) | $12.15 |
| Adafruit PiTFT Cap Touchscreen | B00XW2OI6A | [Amazon](https://tinyurl.com/ybco5so3) | $68.20 |
| Adafruit Faceplate & Buttons | B019MGASTA | [Amazon](https://tinyurl.com/y8yd353a) | $29.98 |
| SMRaza Electronic Starter Kit | B01HRR7EBG | [Amazon](https://tinyurl.com/ycb737cn) | $58.08 |
| Wireless Keyboard | B0173QNVT0 | [Amazon](https://tinyurl.com/ydh8q29g) | $20.00 |
| Parts Kits | N/A | Humber Book Store | $120.00 |
| Total price |  |  | $430.00 |

The budget for my project was $150 dollars, but after I bought everything and got it shipped to my house the total came out to be a bit over $400 as this includes the parts kits we are required to buy when we started this program.

* The SMRaza Electronic Starter kit includes lots of accessories such as a breadboard, wires, LED's and much more but that stuff isn't required for the project.

# 3.0.3 Time Commitment

**Total estimated time**: project still needs to be completed in order to give a completion time

# 3.0.4 Mechanical Assembly

As for the assembly for the touchscreen, it doesn't require any custom PCB or soldering. Install the mini heatsinks by removing the adhesive from the bottom and place both heatsinks on the 2 chips on the Raspberry Pi motherboard. All you are required to do is place the touchscreen on top of the raspberry pi aligning with the pins.

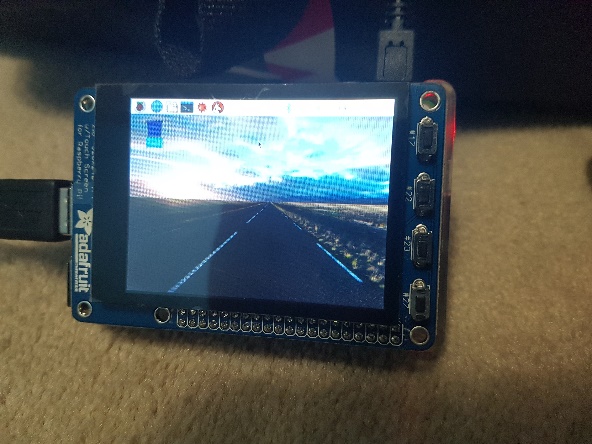
Once mounted and powered up it gives us a white screen, nothing else. To fix this we are required to install the display drivers on the Raspberry Pi. To get the drivers type this in the terminal.

cd ~

wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/adafruit-pitft.sh

chmod +x adafruit-pitft.sh

sudo ./adafruit-pitft.sh

When running these commands you will be asked if you want to use terminal only or GUI only, I chose GUI as you can use the terminal in that but in the terminal, you can’t use the GUI. If you accidentally choose terminal you can revert to GUI by running the script again and choosing the right option. After doing this the screen will look like this.[](https://raw.githubusercontent.com/KaranRajKanwar/DigitalDashboard/master/Blog%20Images/2.8%20inch%20Capacitive%20touchscreen.jpg)

3.0.5 PCB/Soldering

As for the screen soldering isn’t required but for the audio receiver and GPS sensor soldering was required. See below for soldering images.

# 3.0.6 Power Up / Raspberry Pi setup

1. Download Raspbian disc image on a computer via ZIP download or torrent.
2. Unzip the downloaded file using any software of choice, I prefer WinRAR.
3. Write the disc image to the micro SD card (must be a minimum of 8 GB), I prefer Win32 Disk Imager.
4. Plug in your choice of mouse, keyboard, display via HDMI, and power supply.
5. Install Raspbian on the Raspberry Pi by inserting the micro SD card into the slot and giving it power.
6. On first boot you should be taken to a configuration screen, you don't have to do anything here, but I recommend changing your account password for security. If for some reason you were not able to see the configuration screen type the following command in terminal.

sudo raspi-config

1. Update the repositories on the Raspberry Pi by running this command in the terminal.

sudo apt-get update

Sudo apt-get upgrade

# 3.0.7 Case

Still required to create case according to the hardware dimensions

# 3.0.8 Unit Testing

If the touchscreen works and responds correctly with touches, it's confirmed that it’s been installed correctly but to make sure we can check our i2c address by running this command in the terminal.

Sudo i2cdetect -y 1

3.0.9 Android Application Setup

The main requirements for this application is API 28 which is the latest android OS build called Pie. For the most optimum performance experience it is recommended this app be used on a Samsung Galaxy s9 on API 28. There are permissions which are required to be enabled by the android system. This is done when the app if first launched, if you don’t enable these permissions the app will not function as it is supposed to. The app will not install on anything lower than API 28. To get the app source code and APK go to <https://github.com/KaranRajKanwar/DigitalDashboardFinal/tree/master/Mobile%20Application/DigitalDashboard>.

# 4. Results

# 5. Data

# 6. Analysis

# 7. Conclusion

# 8. Bibliography

# 9. Appendix A: Glossary

# 10. Appendix B: Analysis Models

# 11. Appendix C: Calculations