User Documentation BRS

March 2015

1 General Purpose

BRS or (Bike Remote Sensing) is useful for cyclists who are into hobby electronics. The goal of the software in this repository is to allow users to capture proximity sensor data from an arduino and have it display on their android smart phone. By connecting any android enabled phone to the sensor device one can detect objects that could potentially be in a blindspot on a bike. We hope that this make bicycle commuting safer especially in areas with dense traffic.

2 Requirments & Installation

Android version Kitkat (4.4) minimum requirement. Currently not on google play store therefore must be built through AndroidStudios. To install, get AndroidStudios, clone repository then build then run on device through AndroidStudios. Compile and deploy arduino program through prefered arduino IDE. The arduino file is located in `BRS/Arduino_Files/ping_6_sensor.ino`.Now that software is installed connect arduino to your sensors(wiring schematic found here) and connect arduino to phone.

3 Hardware

The 'sensor device' is constructed using a ArduinoUno connected with 6 HR204 Ping Proximity Sensors. The pin schematic can be reference in figure(?). All adjustments to wiring configuration can be set in the arduino file. The usb port is connect to the phone with a microusb adapter. This supplies the power for the arduino and the medium through which they communicate.

4 Features

BRS contains 3 main features

1. A visual system which displays the distance of each sensor to the foreign object.This display operates in real time, and conveys the approach of an object.
2. A alert system which based on the proximity of the forieng object. It will alert the user with a screen flash.
3. A debug mode to asertain the functionality of the program through a log file. Hopefully this will encourage further developement.

5 Usage

5.1 Opening

After device is connected to your 'sensor device', the android system will display a message to open the BRS application. After opening to the main activity, taping the screen will be enough to start the visual display of sensor data.

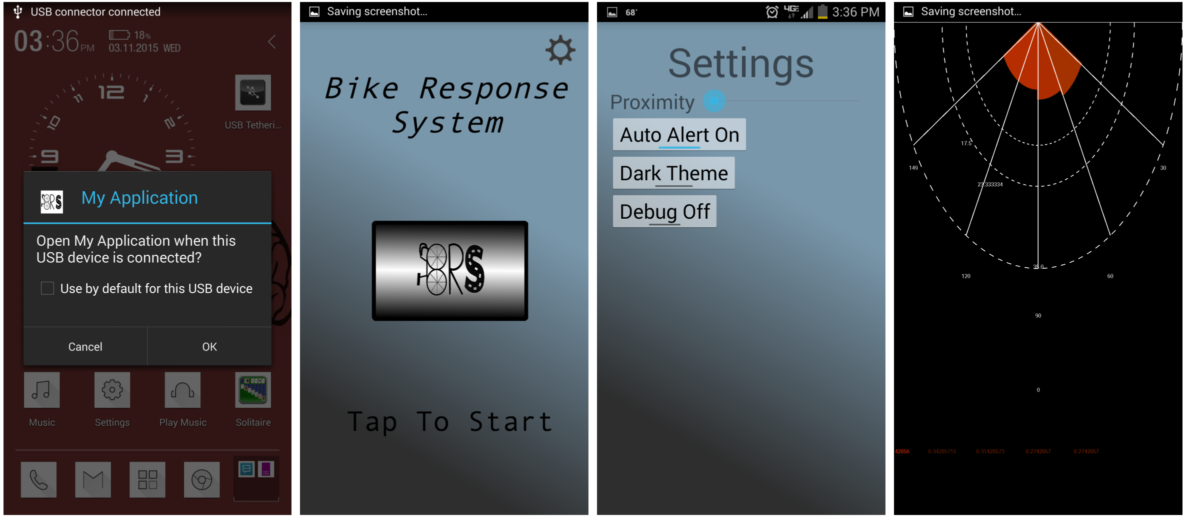
5.2 Settings

Clicking the gear in the top right corner will open a setting page for certain options: Proximity, how sensivity do you want the ALERT system Theme, purely an aesthetic effect ALERT on/off - whether you want the alert system DEBUG on/off - debug the system a (developer option)

5.3 VisualDisplay

The visual display system represents the array of the sensors as a half circle. Each sector represent the field of view of a individual sensor. The distance from a object to the sensor is displayed by using a colored in wedge for that sector. The color and radius of wedge denote the proximity of the object to the user.

Figure 1: Activation, Home Screen, Settings, Visual Display



USB connector connected

My Application

Open My Application when this USB device is connected?

□ Use by default for this USB device

Cancel OK

Saving screenshot...

Bike Response System

Tap To Start

Settings

Proximity

Auto Alert On

Dark Theme

Debug Off

6 Device to Phone Connections

You phone must be connected to the device for the visual display to run. If disconnected it will no longer show any data or return the user to the home screen. If all goes wrong simply close application and replug device this will reset both Android and Arduino executions.

7 Limitation of Sensors

The sensors have a 60 degree field of view. The further the object is from the sensor the more difficult it is for the sensor to capture the reflection from the ultronsonic signal. Therefore it is better at picking large objects like car and less likely to detect smaller ones like people or poles. The maximum range of the sensor used in this project is 10ft.

8 Known Issues

The leftmost sensor will be observably glitchy due to the fact that it is the laster information sent in the usb packet and therefore will be cut off sometimes.