|  |  |
| --- | --- |
| A picture of a winding road and trees  Chainage along the road  PyQGIS in application | Abstract  This document provides details about how chainage along a road can be created and used for developing efficient identification of job locations, especially in rural areas.  Yanjin Wang  Geospatial Programming 17-10-2021 |

Contents

[Developing tool for Chainage Point and Hoisting Data using REFLECT Asset Edge software 2](#_Toc85352086)

[Executive Summary 2](#_Toc85352087)

[Problem definition 2](#_Toc85352088)

[Objective 3](#_Toc85352089)

[Methodology 3](#_Toc85352090)

[Project plan 4](#_Toc85352091)

[The Chainage Breakdown Subroutine 4](#_Toc85352092)

[Application 4](#_Toc85352093)

[Displaying chainage of the tablet 4](#_Toc85352094)

[7](#_Toc85352095)

[Automatic captioning of location of photos 7](#_Toc85352096)

[Conclusion 9](#_Toc85352097)

[References 9](#_Toc85352098)

[Appendix 9](#_Toc85352099)

[Appendix A 10](#_Toc85352100)

[Appendix B 14](#_Toc85352101)

[Figure 1Displaying Chainage using table based on point created using PyQgis example 1 5](#_Toc85352012)

[Figure 2Figure 1Displaying Chainage using table based on point created using PyQgis example 2 6](#_Toc85352013)

[Figure 3 Displaying chainage location and co-ordinate in REFLECT based on dataset created 7](file:///C:\Krishna%20Shrestha\Yanjin%20Study%202021\Geospatial%20programming\Project%20works\Developing%20tool%20for%20Chainage%20Point%20and%20Hoisting%20Data%20using%20REFLECT%20Asset%20Edge%20software%20v0.docx#_Toc85352014)

[Figure 4 Creation of defect in REFLECT software 8](#_Toc85352015)

[Figure 5 Automatic printing of road name, chainage information on the photo example 1 8](#_Toc85352016)

[Figure 6 Automatic printing of road name, chainage information on the photo example 2 9](#_Toc85352017)

# Developing tool for Chainage Point and Hoisting Data using REFLECT Asset Edge software

## Executive Summary

Having a correct dataset is the key for asset management; this will enable to identify and predict locations of asset that need renewal and upgrade. Unfortunatley, a lot of databases, especially in rural Australian council are wrong mostly caused by erroneous measurement.

To eradicate this issue, a tool is produced using PyQGIS which produced chainage along 10 m on councils road. These network of points are then hoisted onto REFLECT application resulting live information of chainage of a point at the tablet dashboard. What’s more, when a photo is taken using the tablet, the chainage point, road name, and coordinate get stamped on the photograph providing accurate information about the asset.

This tool could be used for pegging gravel resheeting, reseal, reconstruction and also exact communication about issues on the asset. For example: if there is a damaged cattle grid on a road, now with a single photo, operations team would know where is the issue which will help in efficient work delivery resulting happier community.

## Problem definition

Most of rural roads in Australia are managed by councils. Every year, council spend huge volume of fund in various tasks including resheeting, resealing, reconstruction etc.

It is important that there is seamless communication between asset officers/asset managers and operations staff to enable work being carried out where it is supposed to be done. Based on my experience, there are many instances where asset manager/engineer schedule works at a particular location, but work is carried out at different location; mostly due to errors in measurement at long distances. Usually, in many rural Australia, people are measuring location of roadworks using car odometer or a meter attached to a car.

Suppose, asset manager wants to do resheeting on a road X from chaninage 15.3 to 18.7 km, chainage starting from intersection of road between X and Y. The operation staff sets zero at the intersection and drives along the road and mark 15.3 km as a start point. This measurement could be wrong in many ways:

1. The error accumulates and it will be difference depending on pressure on tyre.
2. Sometime while the operator drive along the road, he gets call; so he pull his car; and a few hundred m gets error happens this way as well
3. Many a time the operator accidentally goes beyond 15.3 km and he loses references and make a guess and put the peg

Now, work is carried out at wrong location; but it is reported back as being done from 15.3 to 18.7 km. At the end of year, asset engineer records this section of road as brand new (newly resheeted) in his asset register. But, later when he goes to field, he will find that the section doesn’t have enough gravel and need resheeting. This, this problem appear a little, but it has big impact on integrity of asset datasets. Almost identical issues applies for road reseal, microsurfacing, reconstruction etc.

The similar issue arises when staff takes photos of an asset. A field staff goes to site after road is damaged due to flood or bushfire or due to natural asset degradation, and takes several photos. But, after six months, it becomes overwhelming to understand which photo relate to where. There are many occasions, I have experienced that photos were being useless because they were not adequately referenced. Recent years, when natural disaster happens, it is necessary to show the following:

1. What was the condition of asset before natural disaster (flood, fire) occurred?
2. What is the condition after the natural disaster at the same location?
3. Based on these photographic evidences, then council can make claim of damage to state or federal government.

The problem is that there are several, at times thousands of photos are taken, but it is difficult to comprehend these photos without proper information about where the photo was taken.

## Objective

The objective of this project is to overcome the issues faced in the problem statement above. If we could put points along the road at interval of 10 m, and provide some attributes to these points as road and chainage value from a particular origin, then these information can be used into various platform e.g ArcGIS map hosting or other platform e.g. REFLECT. Thus, staff can accurately locate where the start point and end point of particular type of works. Also, the same information can be used in some API’s e.g. REFLECT, which can do stamp on photo based on attributes in GIS table.

## Methodology

To have above objective accomplished, the following methodology is planned:

Step 1: For all roads with same road name, produce features with no topological errors, i.e. all verticies are perfectly snapped; and with no undershoot and overshoot.

Step 2: For all these roads, road name is unique, however on each road there are multiple feature depicting different asset IDs. Ensure that for each asset ID there is single feature exists. If there are multiple feature of same asset ID, join them as a single feature.

Step 3: Develop a tool that can put point along the road at 10 m interval. The tool shall be such that it retains the asset ID as there will be different asset ID for the same road at different chainage point.

Step 4: Breakdown chainage for entire assets with the above method; and and make a single layer of entire road assets chainage point

Step 5: Load these chainage point onto REFLECT software asset Edge

Step 6: Use those chainage tool for :

* Marking job sites for resheeting works
* Marking job sites for microsurfacing and reseal works
* Use those information to communicate staff about actual location where there is issue on the road. Each 10 m chainage point acts as a unique address which can be used many different ways to improve asset data capturing.

## Project plan

The following is the project plan:

1. Firstly note that each road composed on several segments. It is necessary to create a single polylines of entire road of particular attribute e.g. Road X. This can be achieved by joining all road segments and making a single polyline.
2. We will have co-ordinate of first vertex and last vertex. We have to figure out to put points at a specified interval along the traverse of the polyline. This will be hardest part of this project.
3. We have to ensure that each points have attributes table specifying i) origin ii) the co-oridnate iii) chainage iv) road name v) segment no
4. Repeat this for entire of road features
5. Combine all of them in one layer
6. Then publish this information manually either in ArcGIS or other platform
7. Research a bit more about how some API’s can be used to capture the attribute values of the chainage in photo caption etc.

## The Chainage Breakdown Subroutine

The details of the chainage breakdown subroutine flowchart is presented in Appendix B of this document. Associated programs in Pyqgis is presented in Appendix A. For better readability of this document, the details of this subroutine is not described repeatedly rather referred to appendixes.

## Application

### Displaying chainage of the tablet

Now, as shown below, with the REFLECT app that has chainage attributes loaded, the live location of the tablet can be displayed as shown in figure below.

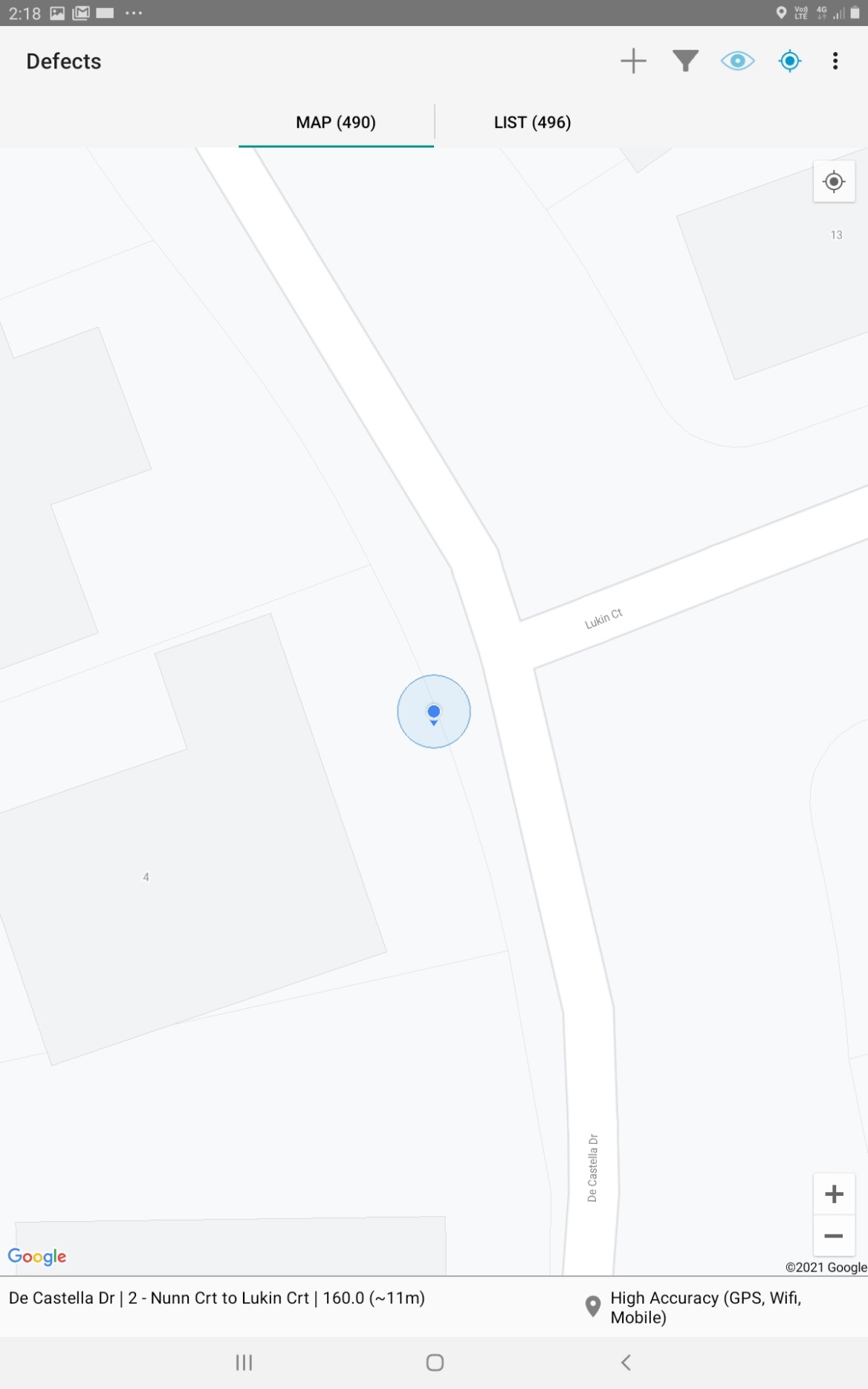


Figure 1Displaying Chainage using table based on point created using PyQgis example 1

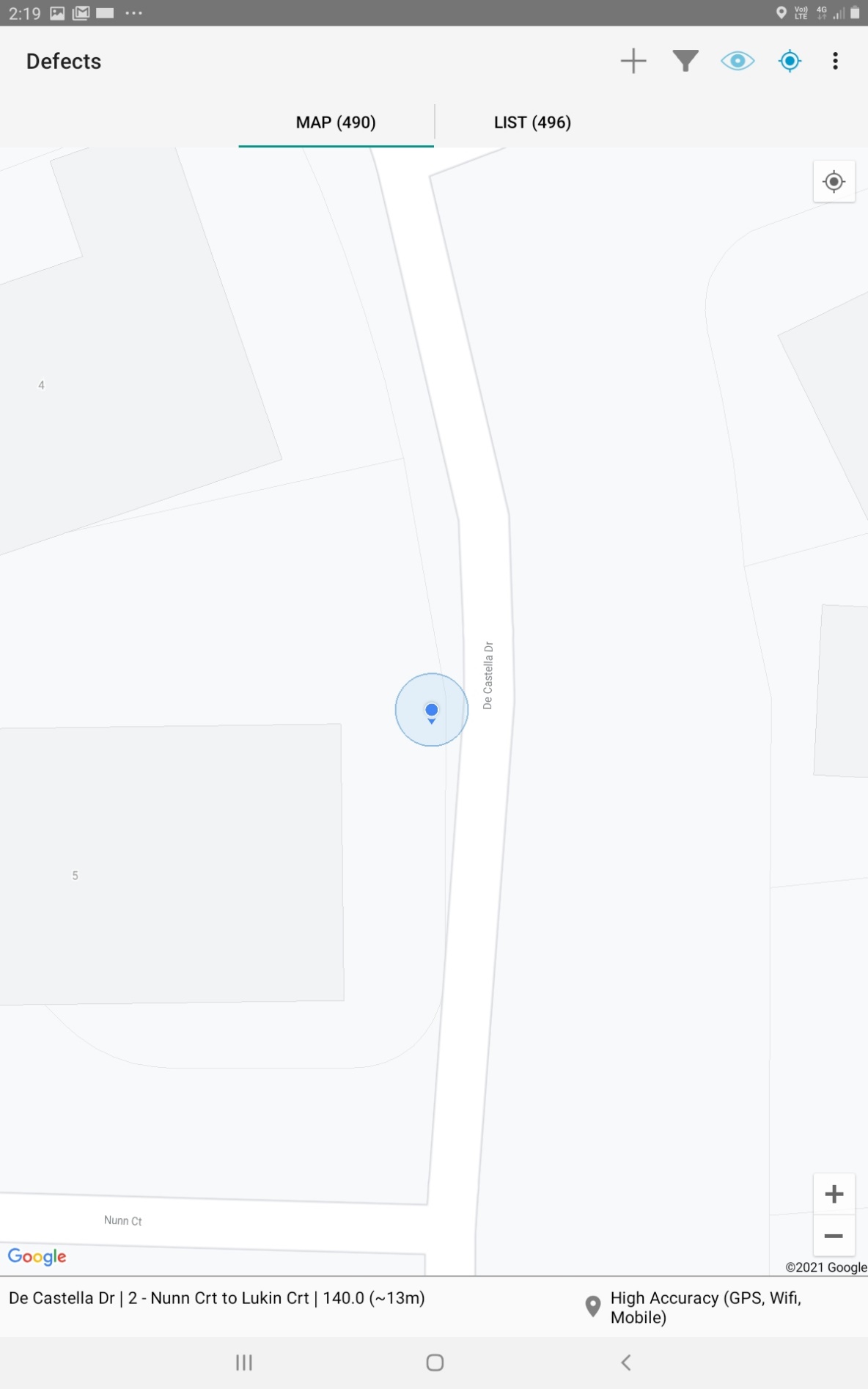


Figure 2Figure 1Displaying Chainage using table based on point created using PyQgis example 2

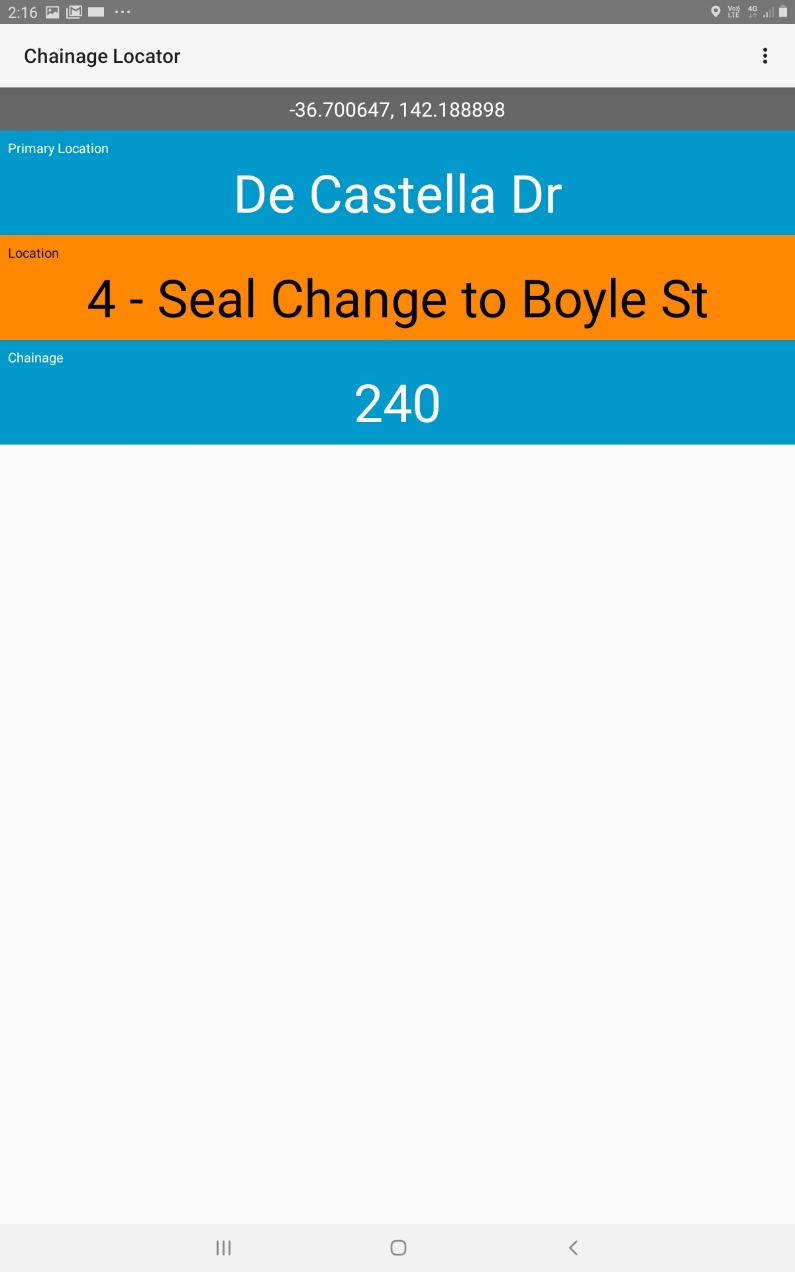
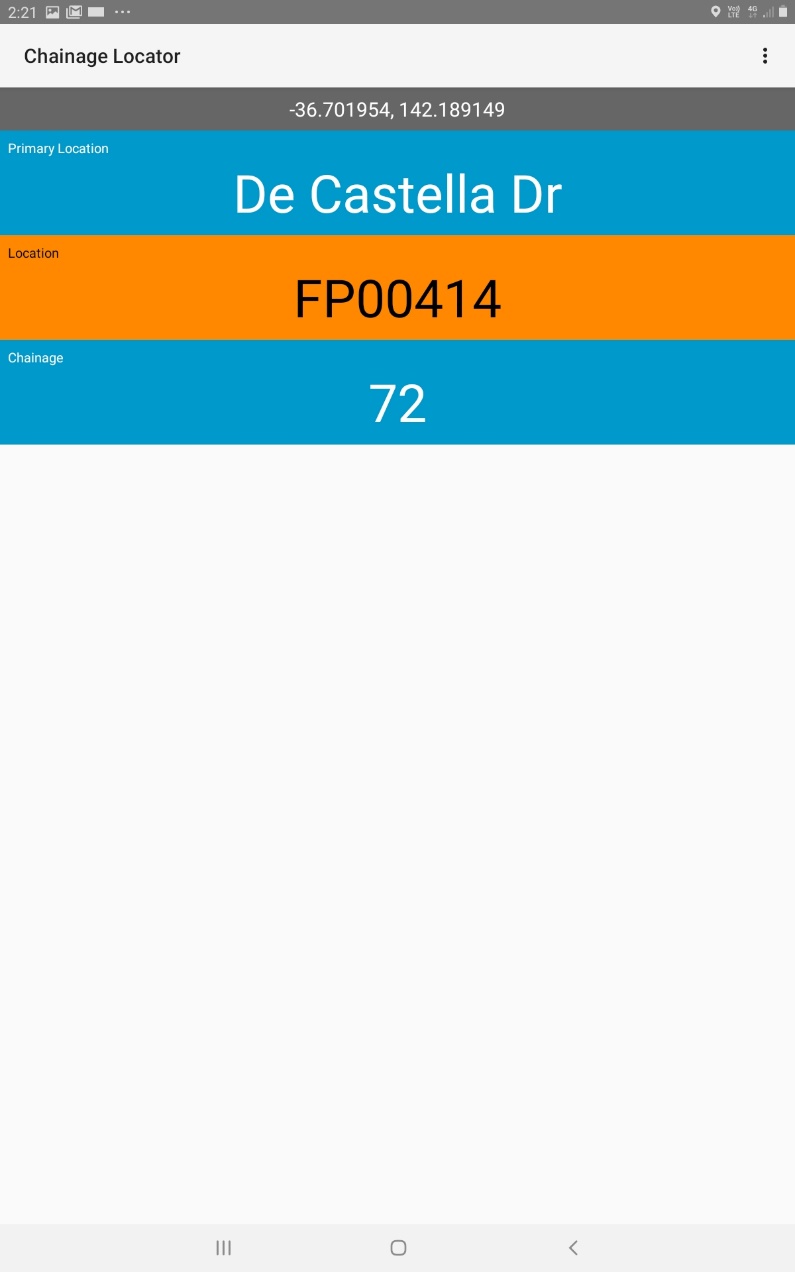


Figure Displaying chainage location and co-ordinate in REFLECT based on dataset created

### Automatic captioning of location of photos

As delineated below, any photo taken using the tablet device can now have chainage location stamp on the photo. This helps greatly to understand where the photo was taken, especially when it is taken at non urban area.

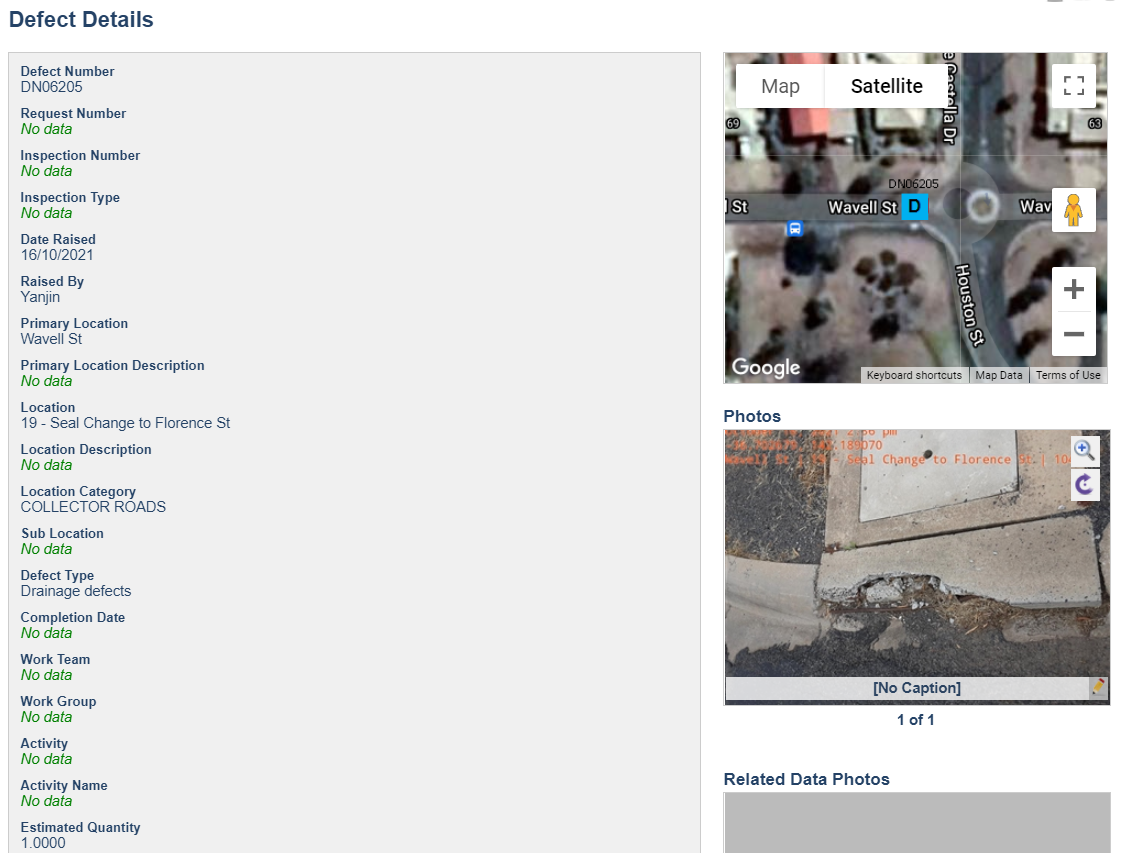


Figure 4 Creation of defect in REFLECT software



Figure 5 Automatic printing of road name, chainage information on the photo example 1



Figure 6 Automatic printing of road name, chainage information on the photo example 2

## Conclusion

With nearly 4 weeks of extensive work, a program has been developed; subsequently; chainage points of entire road assets are placed on REFLECT software. This tool is going to serve various stakeholders within Horsham council including council’s operations staff, road inspector, reseal contractor and also emergency management crew.

## References

[1]*Youtube.com*, 2021. [Online]. Available: https://www.youtube.com/watch?v=3PQiwTICGqs. [Accessed: 16- Oct- 2021]

[2]*Youtube.com*, 2021. [Online]. Available: https://www.youtube.com/watch?v=3PQiwTICGqs. [Accessed: 16- Oct- 2021]

[3]"PyQGIS Developer Cookbook — QGIS Documentation documentation", *Docs.qgis.org*, 2021. [Online]. Available: https://docs.qgis.org/3.16/en/docs/pyqgis\_developer\_cookbook/index.html. [Accessed: 16- Oct- 2021]

## Appendix

### Appendix A

#### Coding of the chainage breakdown program

'''

File: putting\_chainage\_point\_on\_multiline.py

Project: Chainage breakdown

Created Date: 16-10-2021

Author: Yanjin Wang

-----

Last Modified: 16-10-2021

Modified By: Yanjin Wang

-----

Copyright (c) 2021 - Yanjin Wang- AXO Consulting

-----

Program details:

This program produces sets of points at 10 m interval along the feature

Requisite:

1. There should be Asset\_ID field with integer value on the table that has road datasets;

2. There should be Road\_Name field with string value on the table that has road datasets;

3. There should be from field on the table that has road datasets;

4. The dataset should be in EPSG:28354 projectionAcronym

5. Sometime the direction of verticies could be opposite; it is important that the multiline is

swapped before using this tool.argparse

6. Sometime, the line are not fully connected; it is important that line segments are fully connected

Editability:

1. With minimal knowledge of Pyqgis, the above requirements can be modified. For example:

- If the data is on another projection, and location is in different zone, this can be changed. However,

all projections should be in cartesian co-oridnates

- If the road datasets are structured differently; the program can be edited

- Instead of 10 m chainage, if we would like to procude 5 m chainage or 20 m chainage, all '10' in the program

need to be edited to that figure.

Output file:

- The program will produce output file called 'output'. There will be following attributes on the output file

a) Asset\_ID ( it will be taken from original multiline layer)

b) Road\_Name ( it will be taken from original multiline layer)

c) Chainage in m

d) Primary\_Location ( at present it will keep the Road Name as primary location)

e) Location ( at present it will keep the 'from' information from the original layer)

---------- --- ---------------------------------------------------------

'''

from qgis.core import \*

from qgis.gui import \*

import processing

#input the location of the vector file in the following path or edit

infn2="C:\Yanjin\Mod04AssessedEx\Mod04AssessedEx\\road\_renamed.shp"

vlayer1= iface.addVectorLayer(infn2,'','ogr')

#Extracting Asset\_ID, Road\_Name, from information from the input layer

# create fields for the output table

layerFields = QgsFields()

layerFields.append(QgsField('Road\_Name', QVariant.String))

layerFields.append(QgsField('Chainage', QVariant.Double))

layerFields.append(QgsField('Primary\_Location', QVariant.String))

layerFields.append(QgsField('Asset\_ID', QVariant.Int))

layerFields.append(QgsField('Location', QVariant.String))

# create file name for output

fn = 'C:\Yanjin\Module3PracExv2\Module3PracEx\output4.shp'

writer = QgsVectorFileWriter(fn, 'UTF-8', layerFields,\

QgsWkbTypes.Point,\

QgsCoordinateReferenceSystem('EPSG:28354'),\

'ESRI Shapefile')

feat =[]

feat = QgsFeature()

iter = vlayer1.getFeatures()

e=0

for feature in iter:

AssetID=feature['Asset\_ID']

PrimaryLocation=feature['Road\_Name']

RoadName=feature['Road\_Name']

Location=feature['from']

geom=feature.geometry()

length=geom.length()

n=int(length/10)

d=length-n\*10 # checking if the total length is exact multiple of 10

if d==0:

n=n # the iteraction cycle is 1 less if exact multiple of 10

else:

n=n+1

for j in range (n):

c=length-j\*10

if c>=10:

point=geom.interpolate(10\*j)

# Writing data of chainage onto the output file

feat.setGeometry(point)

feat.setAttributes([RoadName,e+10\*j,PrimaryLocation,AssetID,Location])

writer.addFeature(feat)

else: # for last portion which is less than 10 m

point=geom.interpolate(length)

feat.setGeometry(point)

feat.setAttributes([RoadName,e+length,PrimaryLocation,AssetID,Location])

writer.addFeature(feat)

e=e+length

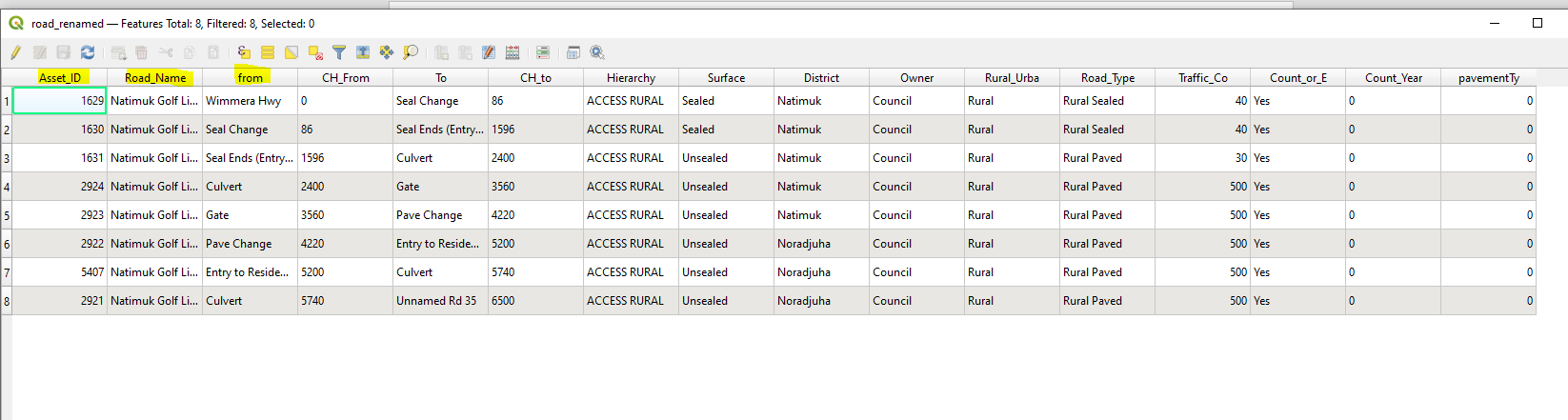
# output file displaying on canvas

layer = iface.addVectorLayer(fn, '', 'ogr')

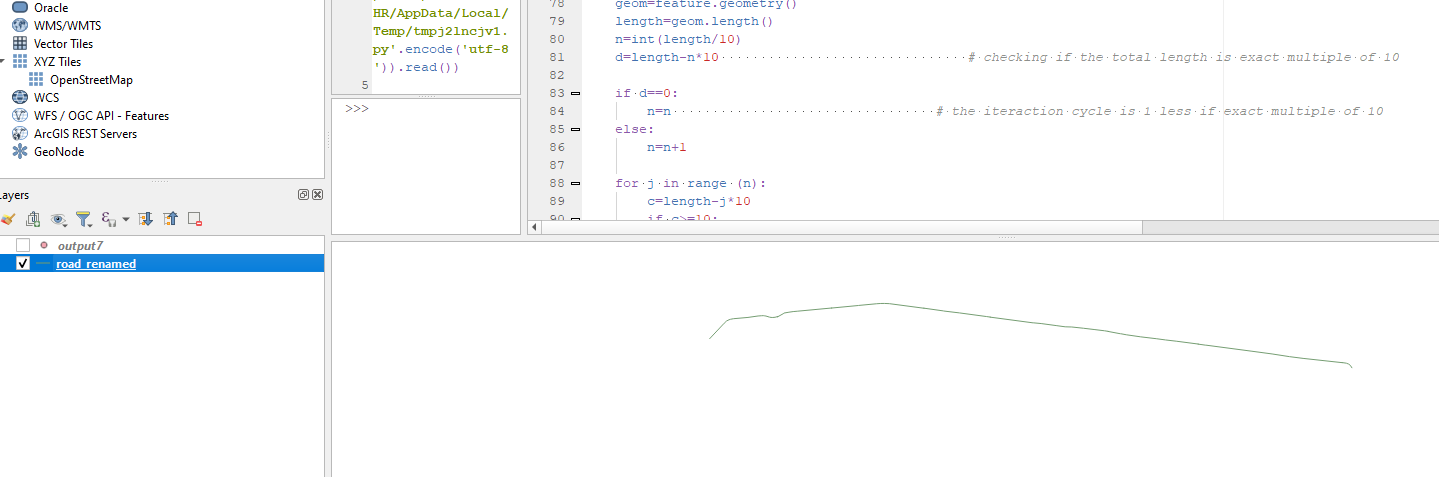
del(writer) # removing the writer

### Appendix B

Input file attribute; the highlighted one are essential

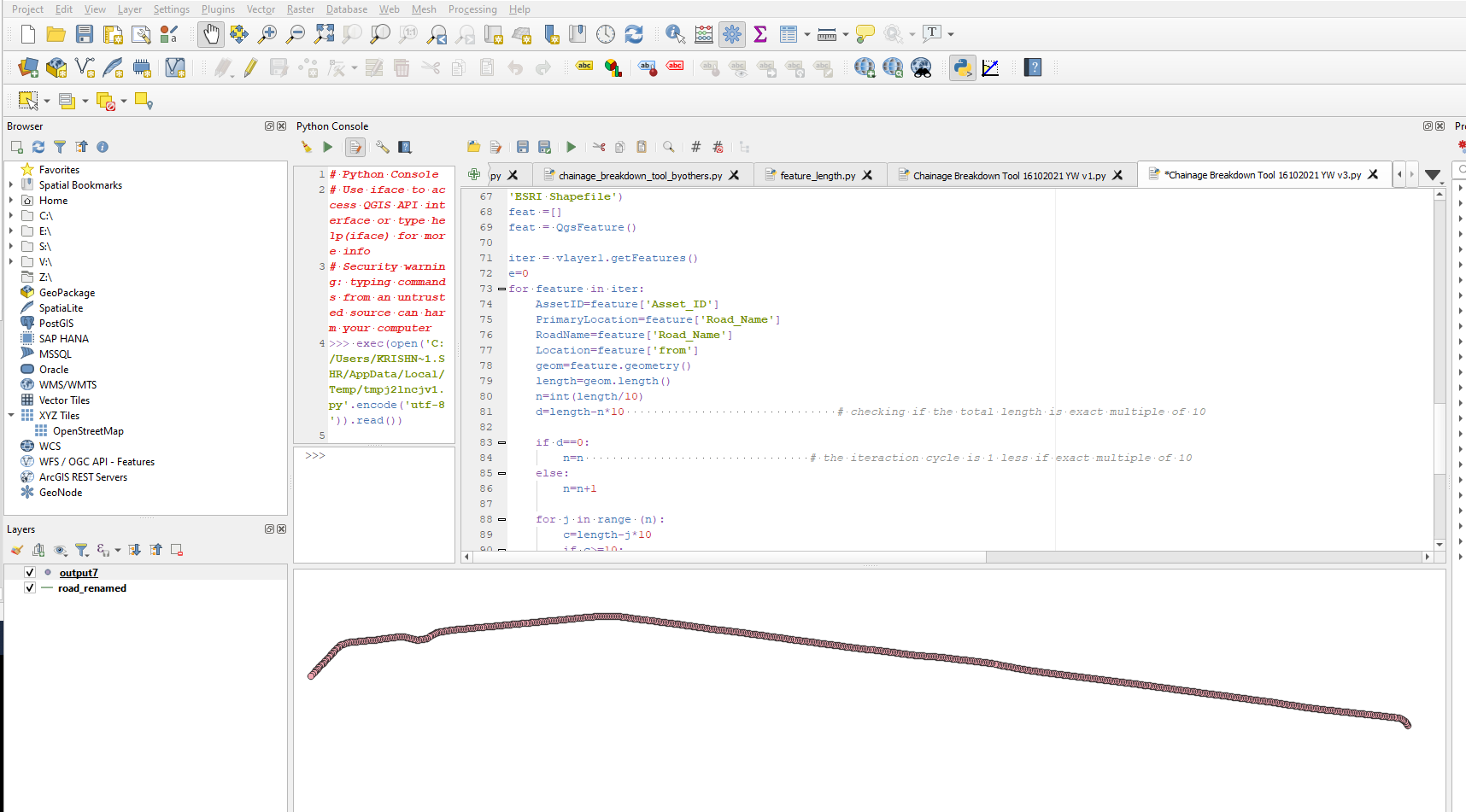


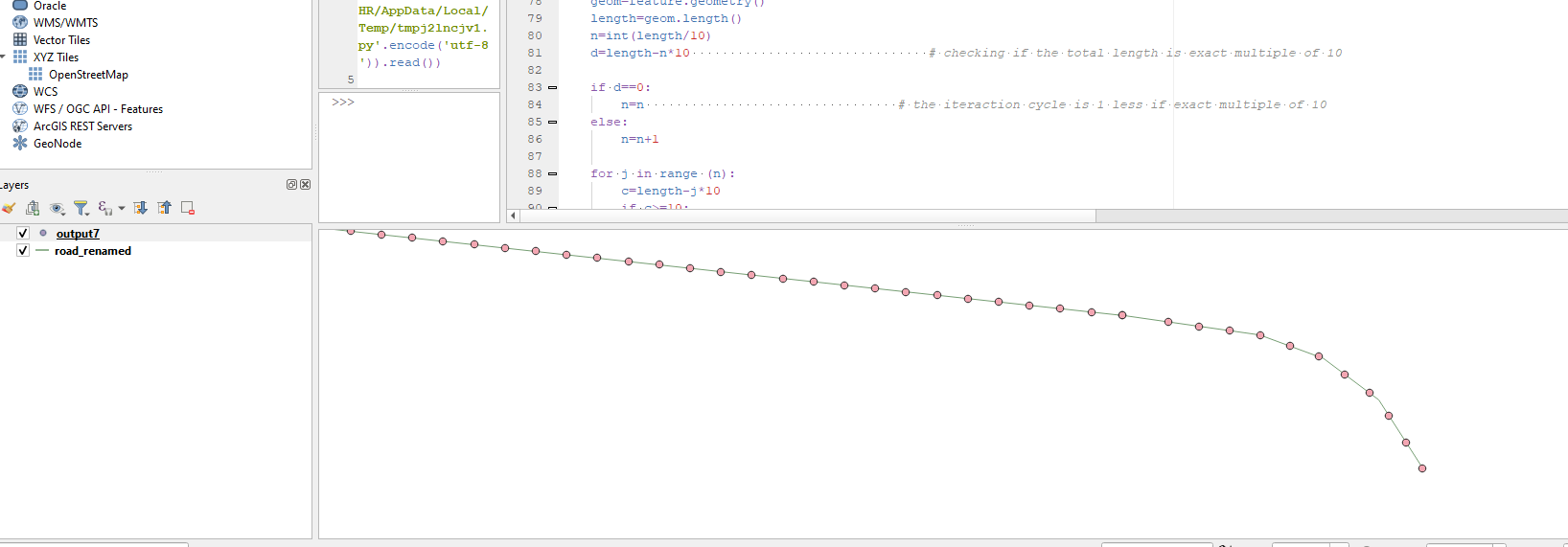
Input file geometry



After running the program:

Output on sample file





### Appendix C

