Driver-Assistant Software Instruction

Jingwen Mo

Yiming Chen

Supervised by Prof. Quoc-Viet Dang

July 23th, 2018

1. Introduction

1.1 Purpose

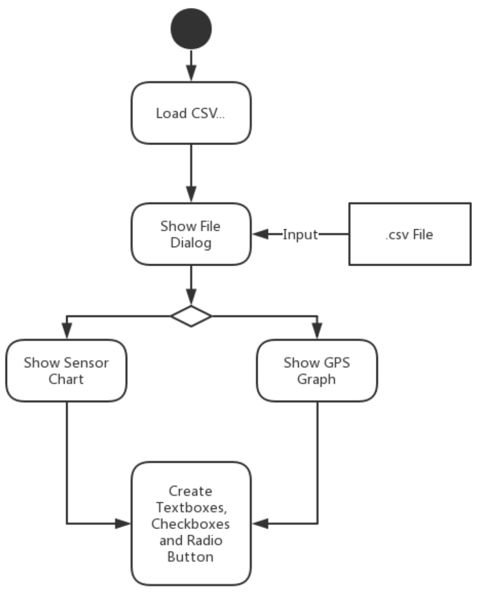
The Data-Graph Software is a PC based software developed to draw the graph which depicts different sensors for the racing car and show the GPS information. This instruction is for the software users and software tester.

1.2 Background

Racing car is one of the fiercest activities. It requires people to have access to different sensors in the car in order to evaluate the performance of the driver and train the driver by the feedback from the sensors. Our software is meant to help coach to transfer the data of different sensors to intuitive graphs easily.

2. Single Run Form

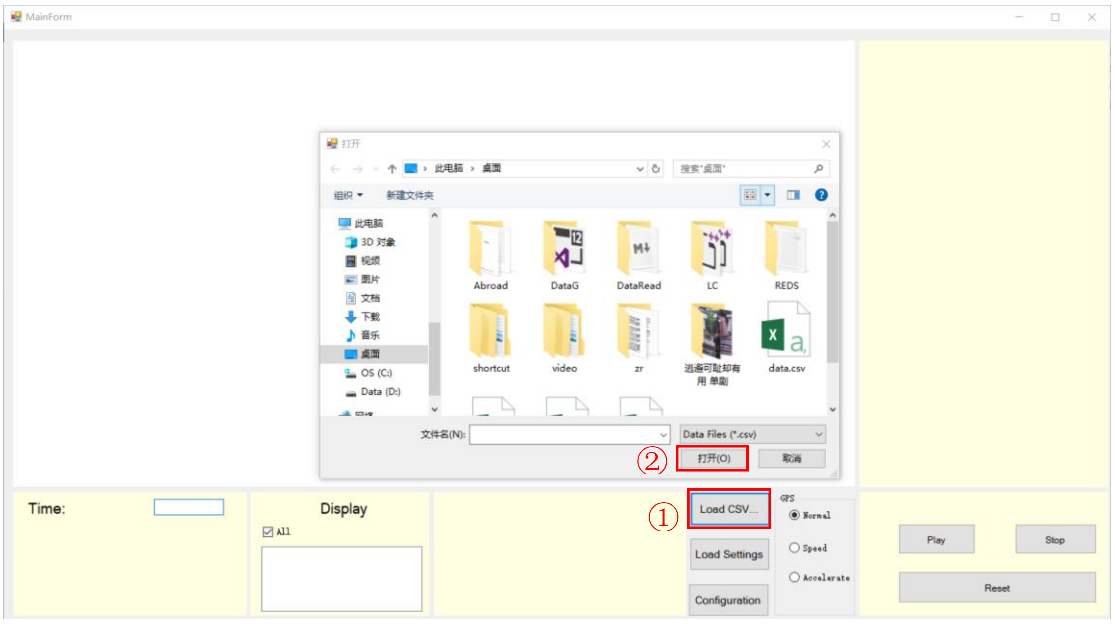
2.1 Load CSV Files

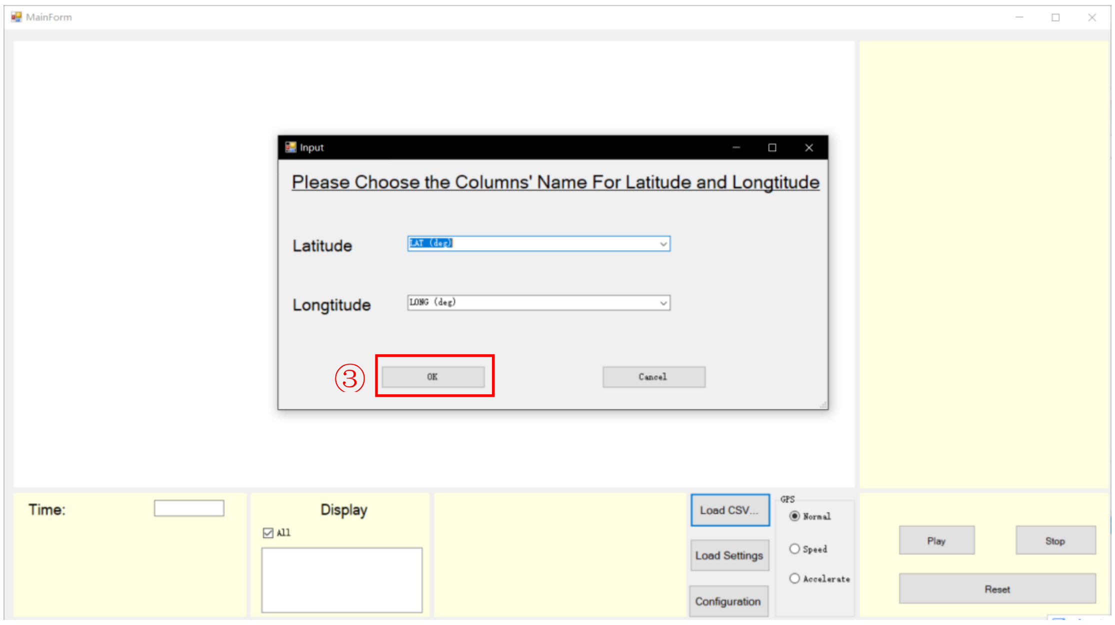


Call function of fileLoadingButton_Click and pop up a file dialog for users to choose their CSV file. After users choose their files and the columns for latitude and longitude, the software could draw graphs for different series of sensors and the GPS graph for the route of driving. Also, the software would dynamic create textboxes,

checkboxes and radio buttons for different sensors read from the CSV file.
Related Functions:

```
void fileLoadingButton_Click(object sender, EventArgs e)
DataTable OpenCSV(string filePath)
```



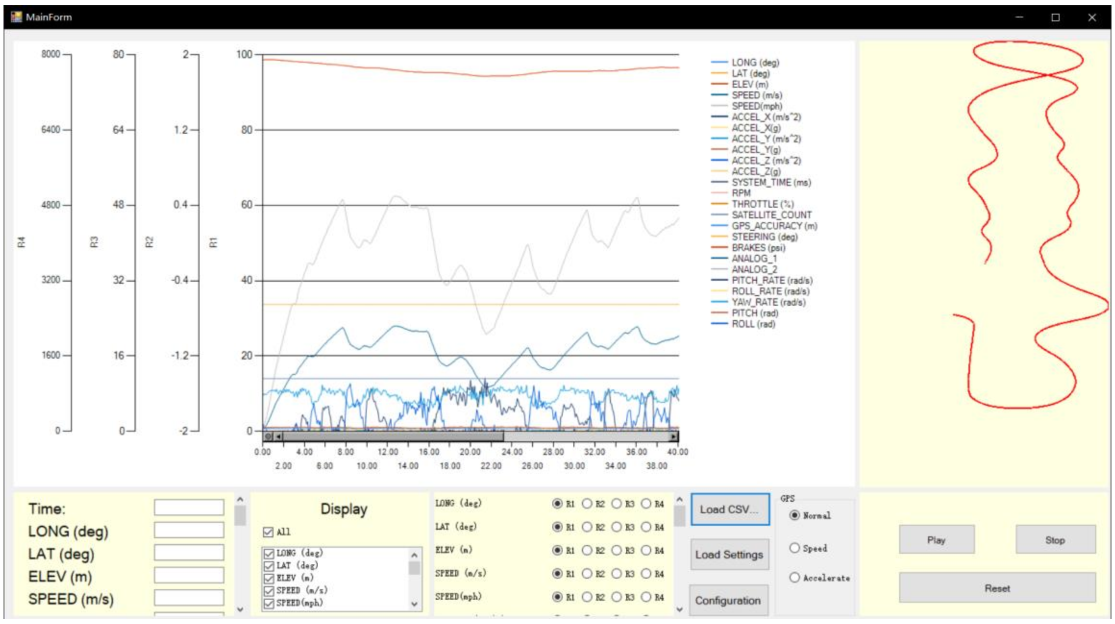


Please Choose the Columns' Name For Latitude and Longtitude

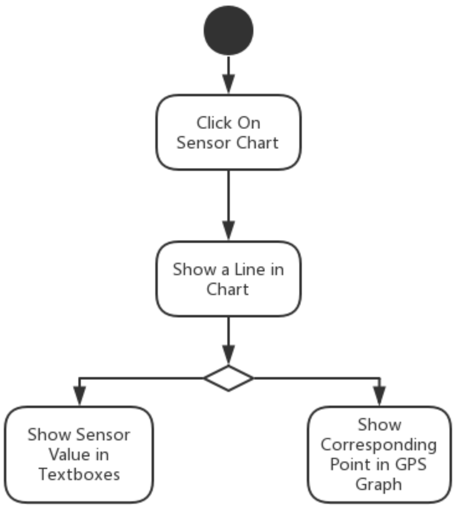
Latitude: LAT (deg)

Longtitude: LONG (deg)

③ OK   Cancel



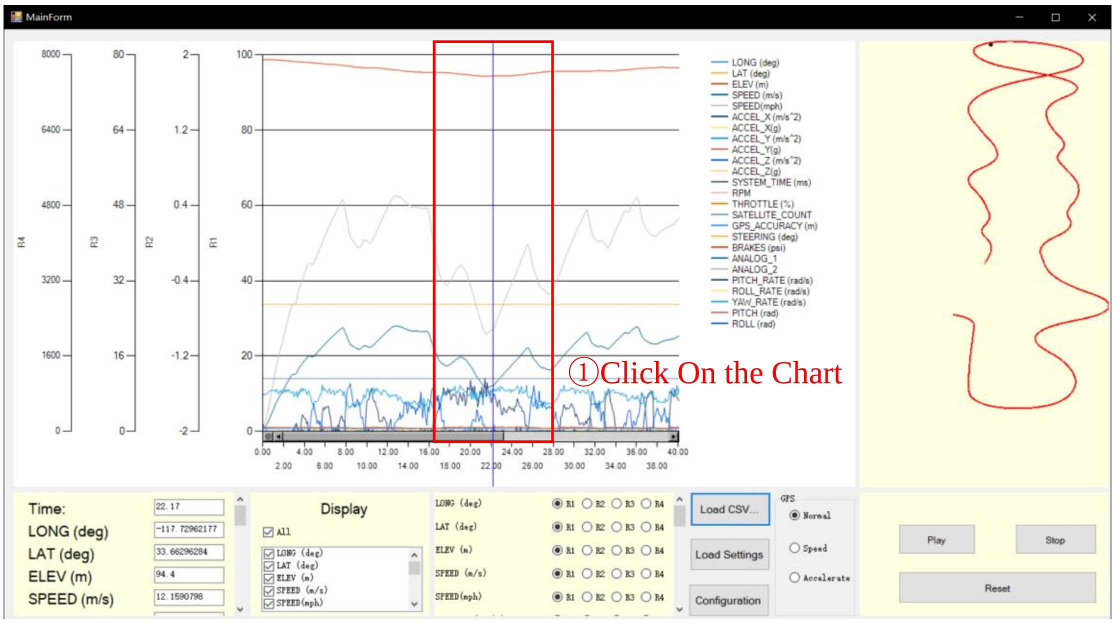
2.2 Show Specific Value of Points


Click On Sensor Chart
Show a Line in Chart
Show Sensor Value in Textboxes
Show Corresponding Point in GPS Graph

Users could click on the graph and show a vertical line at where the mouse clicked. Then the textboxes below would show the values of different sensors. Meanwhile, there would be a black point in the GPS graph to show the same place as you click on the chart.

Related Functions:

```
void sensorChart_MouseClick(object sender, MouseEventArgs e)
int findLeftNear(double value, double[] array, int length)
```



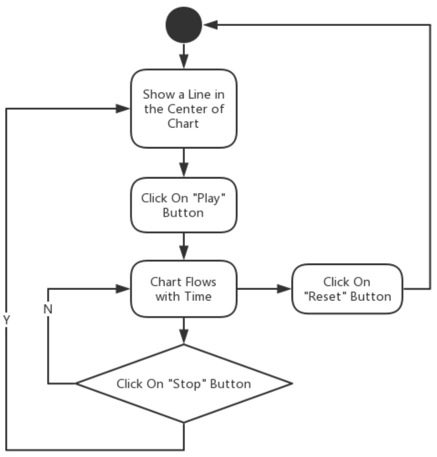
Time: 22.17
LONG (deg) -117.72962177
LAT (deg) 33.66296284
ELEV (m) 94.4
SPEED (m/s) 12.1590798

Display: [✓ All] [✓ LONG (deg)] [✓ LAT (deg)] [✓ ELEV (m)] [✓ SPEED (m/s)] [✓ SPEED(mph)]

LONG (deg) [✓ R1] R2 R3 R4
LAT (deg) [✓ R1] R2 R3 R4
ELEV (m) [✓ R1] R2 R3 R4
SPEED (m/s) [✓ R1] R2 R3 R4
SPEED(mph) [✓ R1] R2 R3 R4

GPS: [✓ Normal] Speed Accelerate

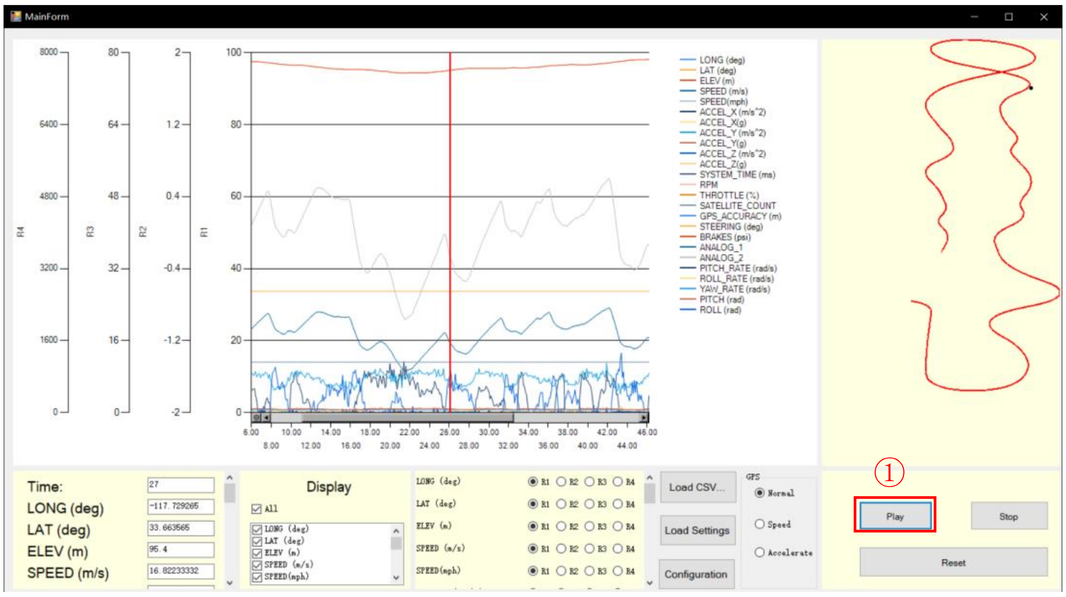
2.3 Replay the Data



Users could click on "Play" button and call the function of buttonPlay_Click. This function enables chartTimer so that it could repeatedly execute the function of chartTimer_Tick at intervals. In the meantime, the textboxes below would show the values of sensors as the chart flowing. When users click on "Stop" button, the chart would stop flowing. After "Play" button is re-clicked, the chart would begin to flow from where it stops. The "Reset" button would let users to reset the chart to the original state.

Related Functions:

void buttonPlay_Click(object sender, EventArgs e)
void buttonStop_Click(object sender, EventArgs e)
void resetButton_Click(object sender, EventArgs e)
void chartTimer_Tick(object sender, EventArgs e)



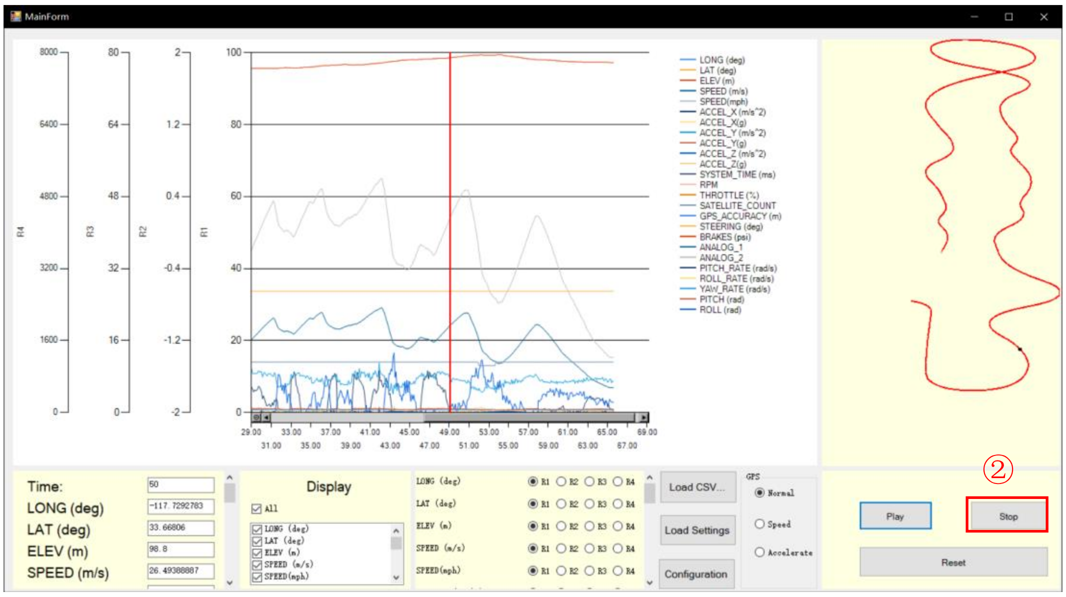
①

Time:	27
LONG (deg)	-117.729265
LAT (deg)	33.663565
ELEV (m)	95.4
SPEED (m/s)	16.82233332

Display: [✓ All] [✓ LONG (deg)] [✓ LAT (deg)] [✓ ELEV (m)] [✓ SPEED (m/s)] [✓ SPEED(mph)]

LONG (deg) [✓ R1] R2 R3 R4
LAT (deg) [✓ R1] R2 R3 R4
ELEV (m) [✓ R1] R2 R3 R4
SPEED (m/s) [✓ R1] R2 R3 R4
SPEED(mph) [✓ R1] R2 R3 R4

GPS: [✓ Normal] Speed Accelerate



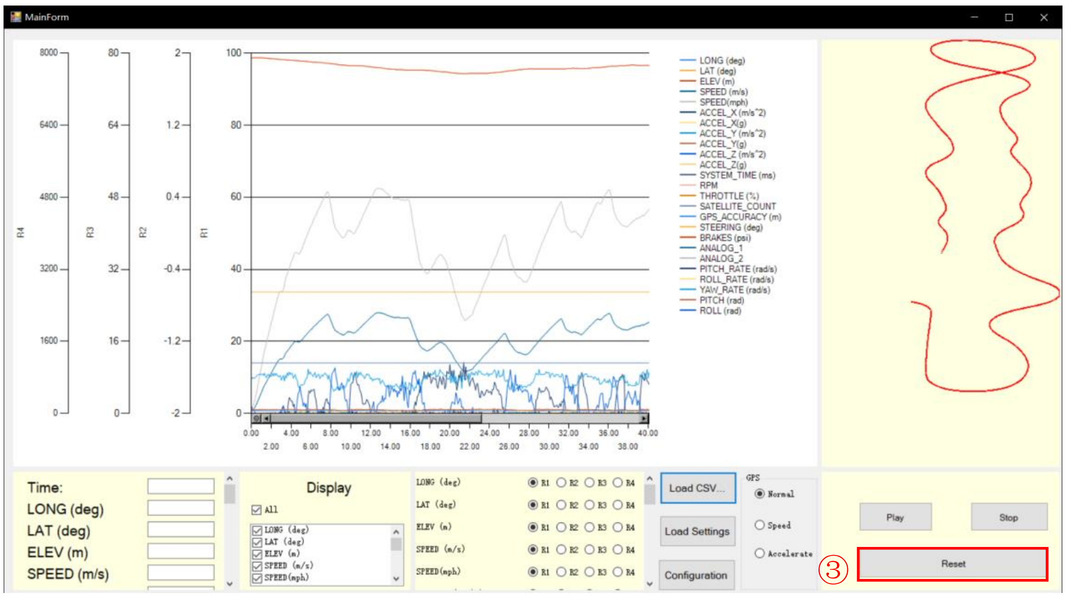
②

Time:	50
LONG (deg)	-117.7292783
LAT (deg)	33.66806
ELEV (m)	98.8
SPEED (m/s)	26.49388887

Display: [✓ All] [✓ LONG (deg)] [✓ LAT (deg)] [✓ ELEV (m)] [✓ SPEED (m/s)] [✓ SPEED(mph)]

LONG (deg) [✓ R1] R2 R3 R4
LAT (deg) [✓ R1] R2 R3 R4
ELEV (m) [✓ R1] R2 R3 R4
SPEED (m/s) [✓ R1] R2 R3 R4
SPEED(mph) [✓ R1] R2 R3 R4

GPS: [✓ Normal] Speed Accelerate



③

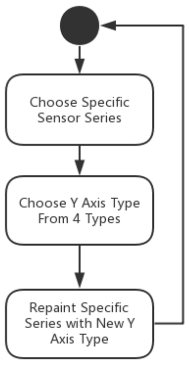
Time:	
LONG (deg)	
LAT (deg)	
ELEV (m)	
SPEED (m/s)	

Display: [✓ All] [✓ LONG (deg)] [✓ LAT (deg)] [✓ ELEV (m)] [✓ SPEED (m/s)] [✓ SPEED(mph)]

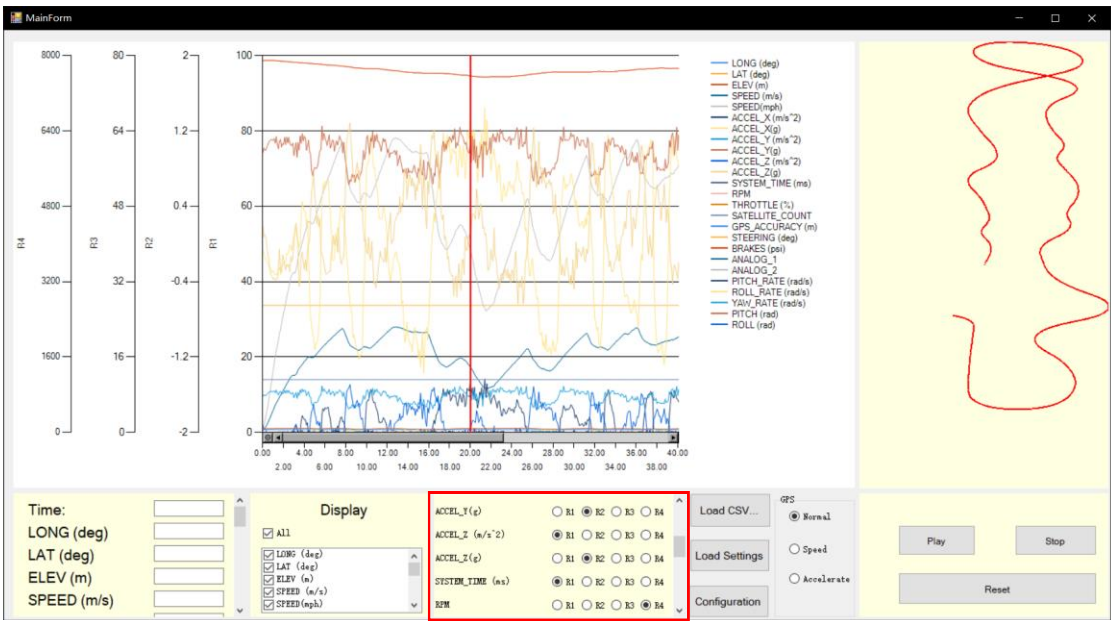
LONG (deg) [✓ R1] R2 R3 R4
LAT (deg) [✓ R1] R2 R3 R4
ELEV (m) [✓ R1] R2 R3 R4
SPEED (m/s) [✓ R1] R2 R3 R4
SPEED(mph) [✓ R1] R2 R3 R4

GPS: [✓ Normal] Speed Accelerate

2.4 Choose Different Type of Y Axis


Choose Specific Sensor Series
Choose Y Axis Type From 4 Types
Repaint Specific Series with New Y Axis Type

Users could change different series from one Y axis to another one. The software provides 4 different Y axes. We can switch between different axes by radio buttons dynamically created by fileLoadingButton_Click function.
Related Functions:
void change(int no, ChartArea caR)
void rb1_Click(object sender, EventArgs e)
void rb2_Click(object sender, EventArgs e)
void rb3_Click(object sender, EventArgs e)
void rb4_Click(object sender, EventArgs e)



Time:
LONG (deg)
LAT (deg)
ELEV (m)
SPEED (m/s)

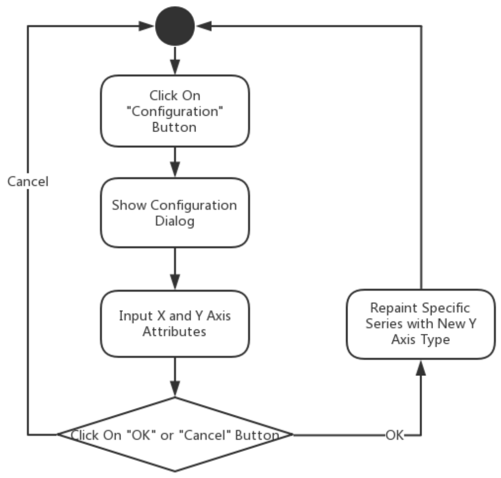
[✓ All]
[✓ LONG (deg)]
[✓ LAT (deg)]
[✓ ELEV (m)]
[✓ SPEED (m/s)]
[✓ SPEED(mph)]

ACCEL_Y(g)	R1	[✓ R2]	R3	R4
ACCEL_Z (m/s^2)	[✓ R1]	R2	R3	R4
ACCEL_Z(g)	R1	[✓ R2]	R3	R4
SYSTEM_TIME (ms)	[✓ R1]	R2	R3	R4
RPM	R1	R2	R3	[✓ R4]

GPS
[✓ Normal] Speed Accelerate

①Choose Different Y Types

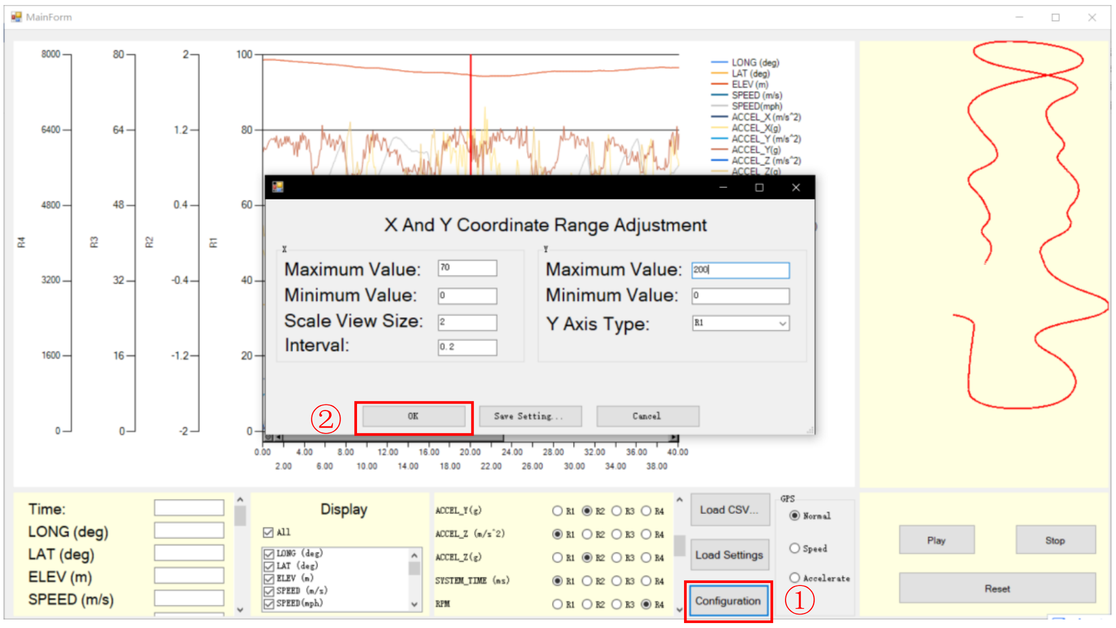
2.5 Customize X axis and Y axis


Click On "Configuration" Button
Show Configuration Dialog
Input X and Y Axis Attributes
Click On "OK" or "Cancel" Button
Cancel
OK
Repaint Specific Series with New Y Axis Type

Users could click on “Configuration” button to pop out a dialog for X axis and Y axis customization. Users could change the range of X axis range, scale and interval and Y axis range and type.

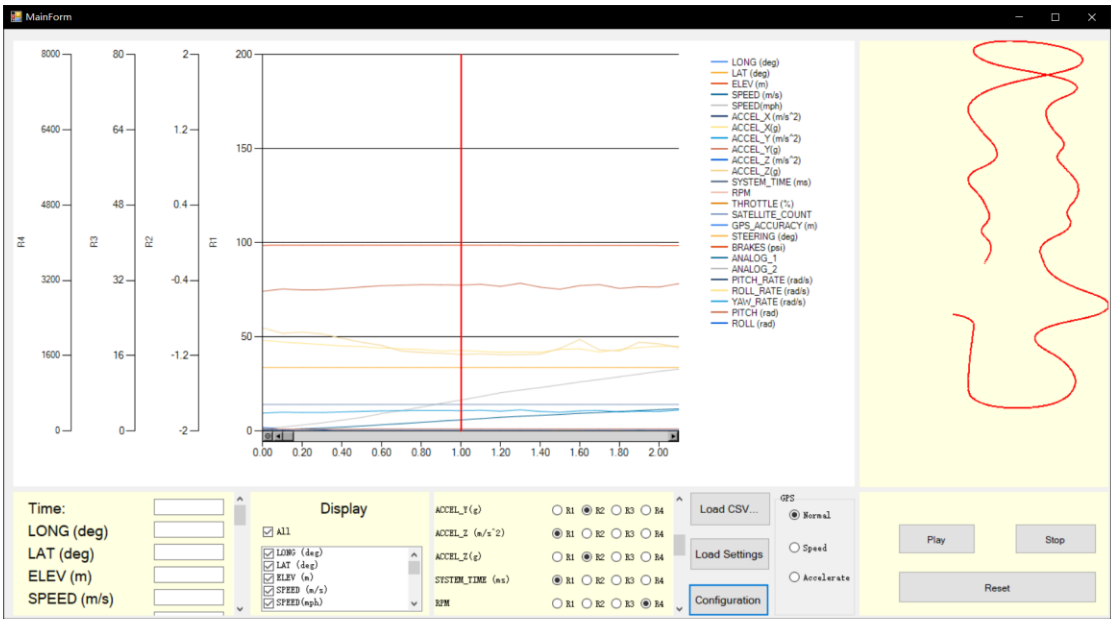
Related Functions and Forms:

```
public partial class RangeForm : Form
void YRangeForm_Load(object sender, EventArgs e)
void confirmButton_Click(object sender, EventArgs e)
void ConfigureButton_Click(object sender, EventArgs e)
```


X And Y Coordinate Range Adjustment

X
Maximum Value: 70
Minimum Value: 0
Scale View Size: 2
Interval: 0.2

Y
Maximum Value: 200
Minimum Value: 0
Y Axis Type: R1



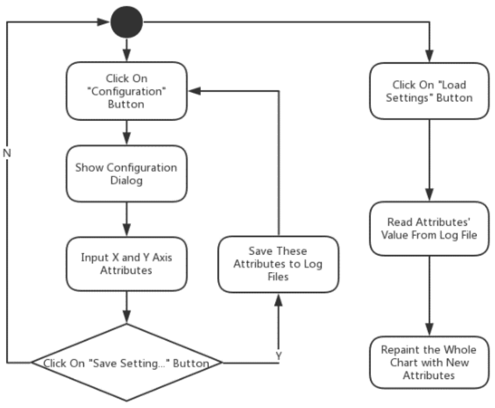
Time:
LONG (deg)
LAT (deg)
ELEV (m)
SPEED (m/s)

Display
[✓ All]
[✓ LONG (deg)]
[✓ LAT (deg)]
[✓ ELEV (m)]
[✓ SPEED (m/s)]
[✓ SPEED(mph)]

ACCEL_Y(g): R1 [✓ R2] R3 R4
ACCEL_Z (m/s^2): [✓ R1] R2 R3 R4
ACCEL_Z(g): R1 [✓ R2] R3 R4
SYSTEM_TIME (ms): [✓ R1] R2 R3 R4
RPM: R1 R2 R3 [✓ R4]

GPS: [✓ Normal] Speed Accelerate

2.6 Save and Load Setting Log Files

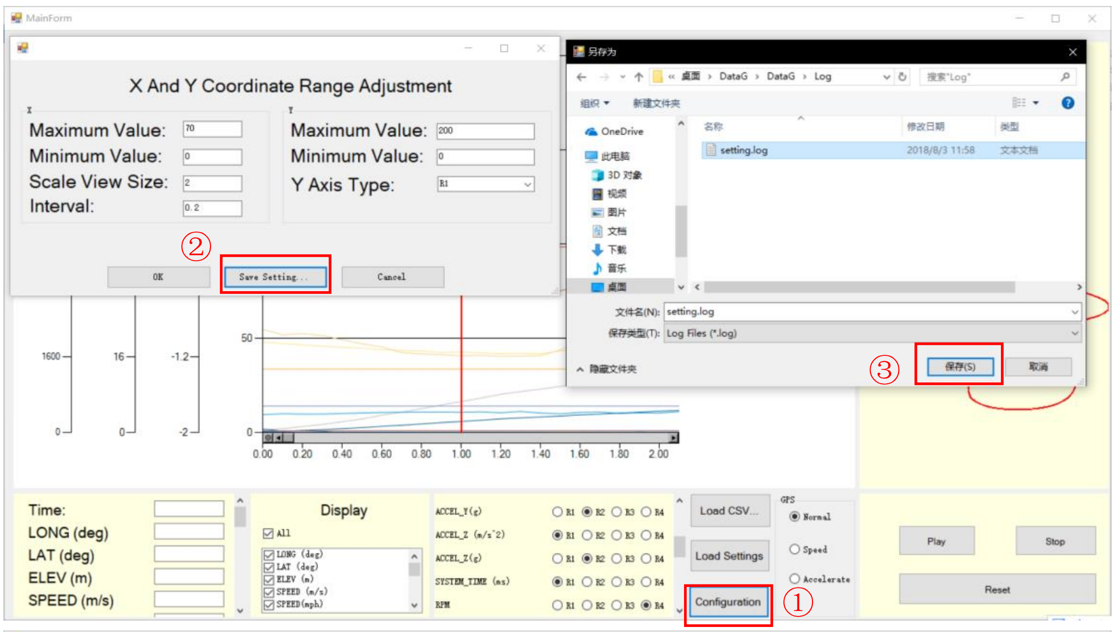

Click On "Configuration" Button
Show Configuration Dialog
Input X and Y Axis Attributes
Click On "Save Setting..." Button
N
Y
Save These Attributes to Log Files
Click On "Load Settings" Button
Read Attributes' Value From Log File
Repaint the Whole Chart with New Attributes

By clicking button of "Saving Setting…" in RangeForm, users could save log files with the X axis and Y axis configuration in specific form. Also, users could load log files created by them by clicking the button of "Loading Settings".

Related Functions:

void settingSaveButton_Click(object sender, EventArgs e)

void settingButton_Click(object sender, EventArgs e)

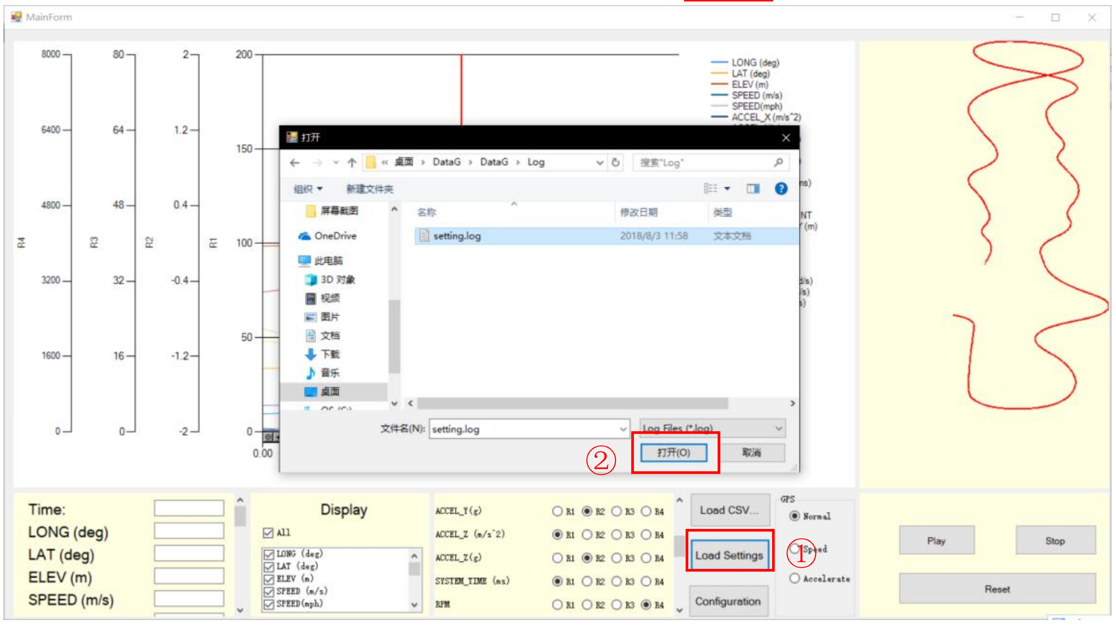

X And Y Coordinate Range Adjustment

X
Maximum Value: 70
Minimum Value: 0
Scale View Size: 2
Interval: 0.2

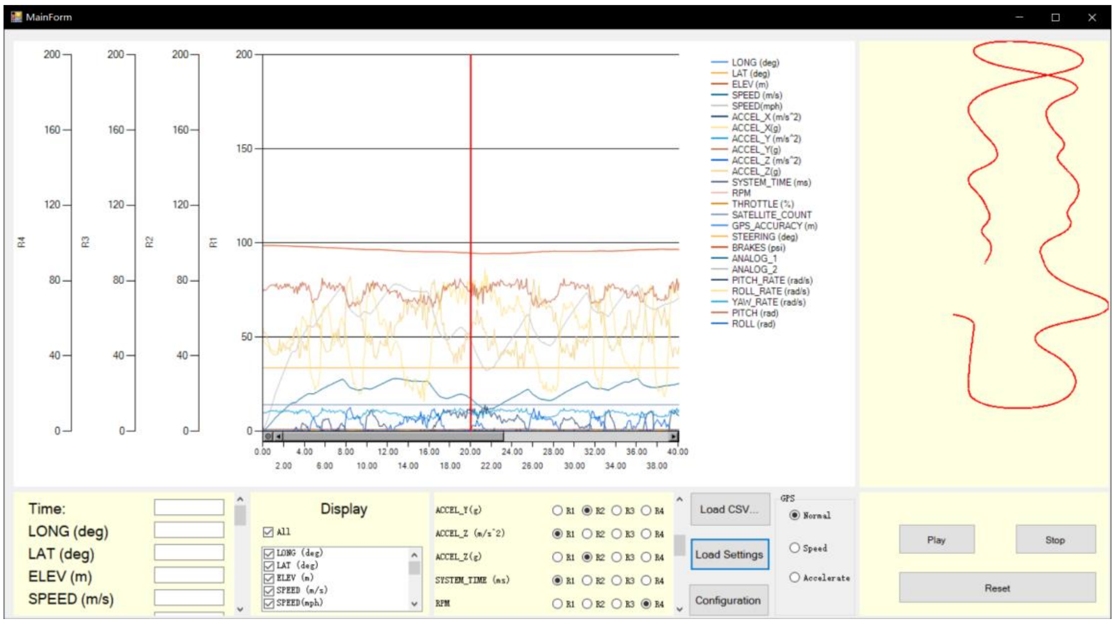
Y
Maximum Value: 200
Minimum Value: 0
Y Axis Type: R1

另存为
文件名(N): setting.log
保存类型(T): Log Files (*.log)

名称	修改日期	类型
setting.log	2018/8/3 11:58	文本文档


打开
文件名(N): setting.log
Log Files (*.log)

名称	修改日期	类型
setting.log	2018/8/3 11:58	文本文档



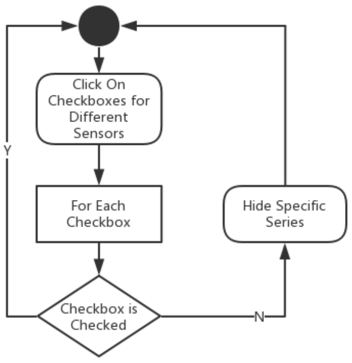
Time:
LONG (deg)
LAT (deg)
ELEV (m)
SPEED (m/s)

Display
[✓] All
[✓] LONG (deg)
[✓] LAT (deg)
[✓] ELEV (m)
[✓] SPEED (m/s)
[✓] SPEED(mph)

ACCEL_Y(g) R1 [✓ R2] R3 R4
ACCEL_Z (m/s^2) [✓ R1] R2 R3 R4
ACCEL_Z(g) R1 [✓ R2] R3 R4
SYSTEM_TIME (ms) [✓ R1] R2 R3 R4
RPM R1 R2 R3 [✓ R4]

GPS
[✓ Normal]
Speed
Accelerate

2.7 Show or Hide Specific Series

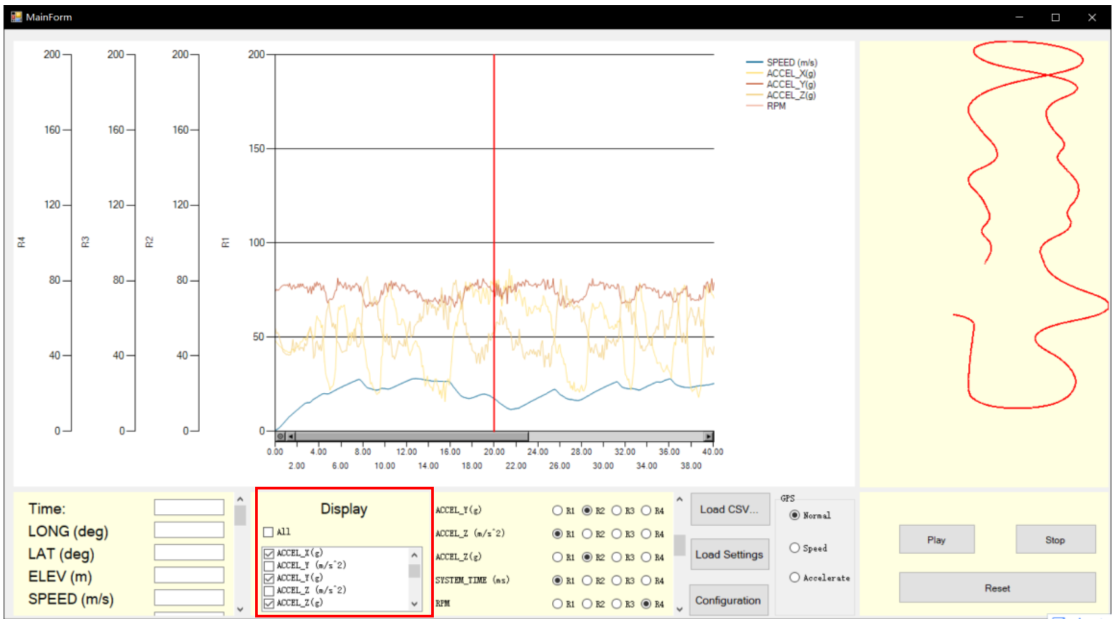


Users could click on different checkboxes created dynamically by fileLoadingButton_Click function. In addition, there is an allSelectedCheckBox for the selection of all.

Related Functions:

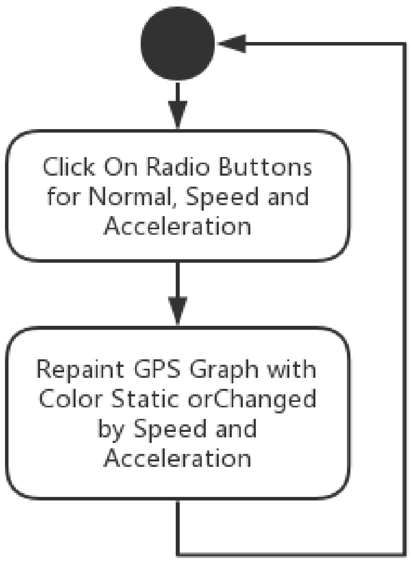
void sensorCheckedListBox_ItemCheck(object sender, ItemCheckEventArgs e)

void allSelectedCheckBox_CheckedChanged(object sender, EventArgs e)



①Click On Series Needed to be Show

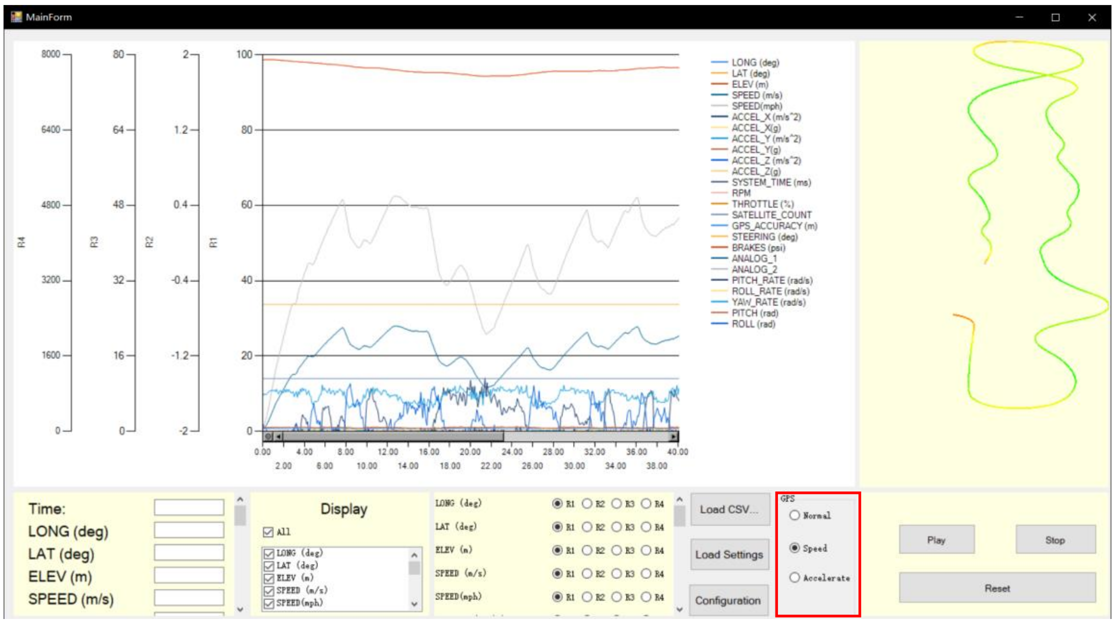
2.8 Show Colored GPS Graph Changed by Speed or Acceleration


Click On Radio Buttons for Normal, Speed and Acceleration
Repaint GPS Graph with Color Static orChanged by Speed and Acceleration

Users could click on "radioButton_Normal" to show normal GPS graph, "radioButton_Speed" to show GPS graph changed by speed where green represents high speed and red represents low speed and "radioButton_Accelerate" to show GPS graph changed by acceleration.
Related Functions:

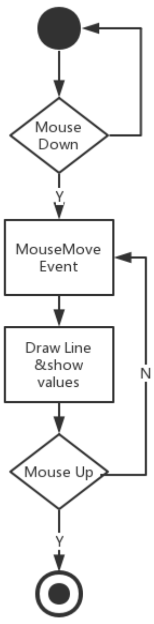
```
void radioButton_Normal_CheckedChanged(object sender, EventArgs e)
```

void radioButton_Speed_CheckedChanged(object sender, EventArgs e)
void radioButton_Accelerate_CheckedChanged(object sender, EventArgs e)
int colorRed(double x)
int colorGreen(double x)



① Click On Normal, Speed or Accelerate Radio Button

2.9 Drag Mouse on the Chart



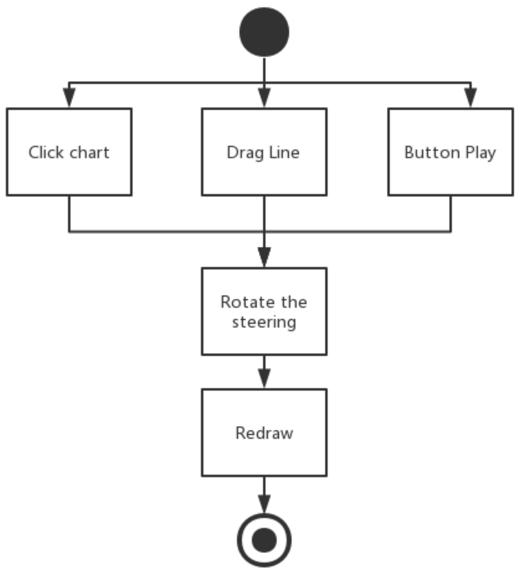
Users can drag the red line randomly in the chart. The chart will listen on "MouseDown" event to trigger "MouseMove" event and "MouseUp" event to end "MouseMove" event.

Related Functions:

void sensorChart_MouseMove(object sender, MouseEventArgs e)

void sensorChart_MouseDown(object sender, MouseEventArgs e)
void sensorChart_MouseUp(object sender, MouseEventArgs e)

2.10 Show Steering Position



When user clicks on the chart, the steering wheel will be rotated a certain angle according to the CSV file. Similarly, if user clicks on Button Play, the angle of the steering wheel will be changed in real time.
Related Functions:
public static Image RotateImage(Image img, float rotationAngle);
private void sensorChart_MouseMove(object sender, MouseEventArgs e);
private void chartTimer_Tick(object sender, EventArgs e);
private void sensorChart_MouseClick(object sender, MouseEventArgs e);

3. Compared Run Form

3.1 Load CSV Files

3.2 Show Specific Value of Points

3.3 Replay the Data

3.4 Choose Different Type of Y Axis

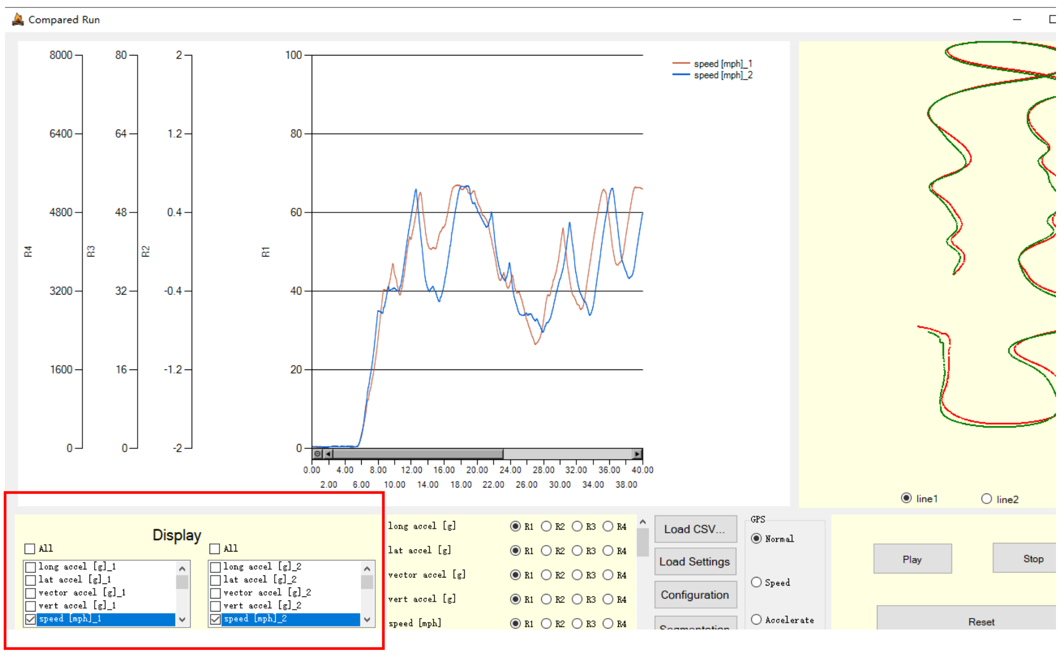
3.5 Customize X axis and Y axis

3.6 Save and Load Setting Log Files

This function is really similar to the single ride. See 2.6.

3.7 Show or Hide Specific Series

There are two checkboxeslists to show the data of two drivers. Users could click on different checkboxes created dynamically when reading the csv file. In addition, there is an allSelectedCheckBox for the selection of all.

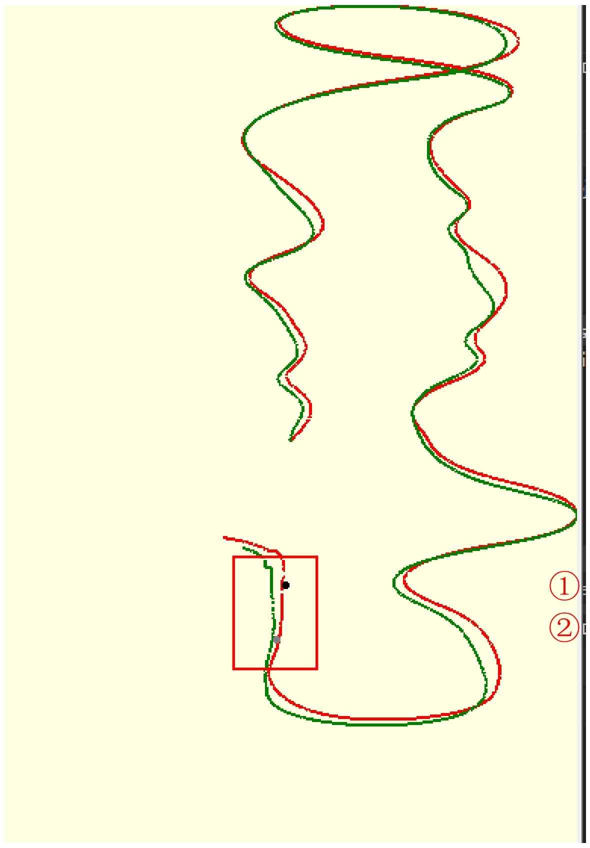


① Click On Series Needed to be Show

Related Functions:
void sensorCheckedListBox_ItemCheck(object sender, ItemCheckEventArgs e)
void allSelectedCheckBox_CheckedChanged(object sender, EventArgs e)

3.8 Click on the GPS map

When we click on one road, it can show where another driver is. With this, we can easily compare the performance of the two drivers.



Related Functions:

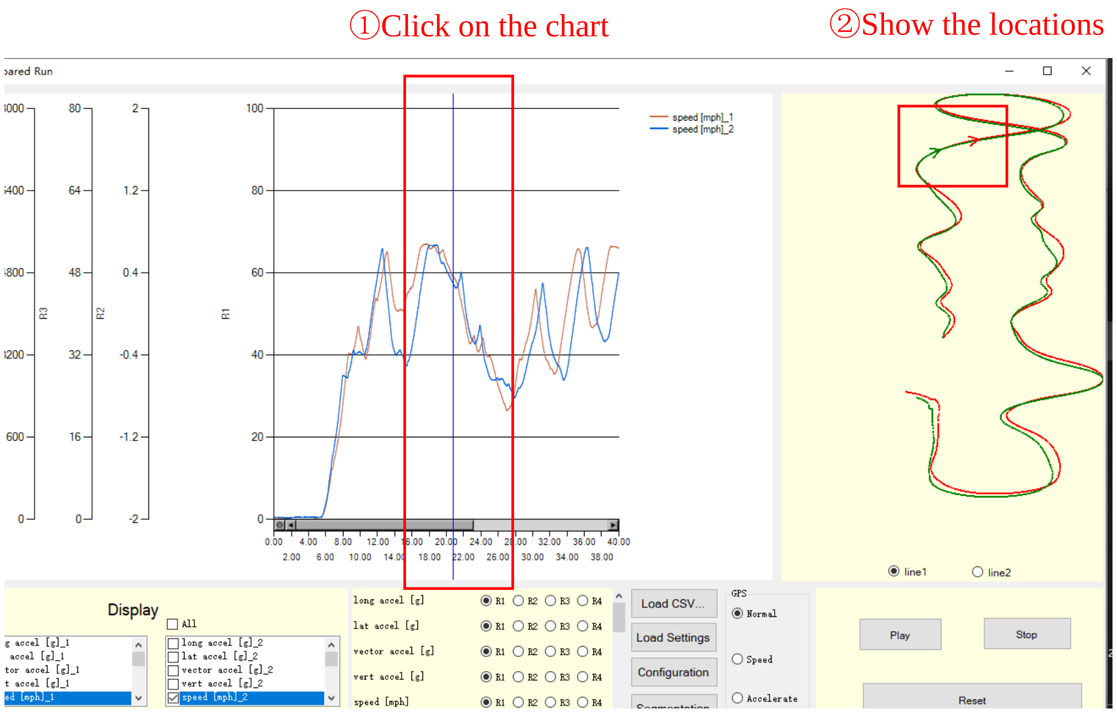
```
private void GPSPanel_MouseClick(object sender, MouseEventArgs e)
```

3.9 Click/Drag Mouse on the Chart

-Users can drag the line randomly in the chart. The chart will listen on “MouseDown” event to trigger “MouseMove” event and “MouseUp” event to end “MouseMove” event.
- Users can click on the chart. It can show the locations & heading position of the two drivers.

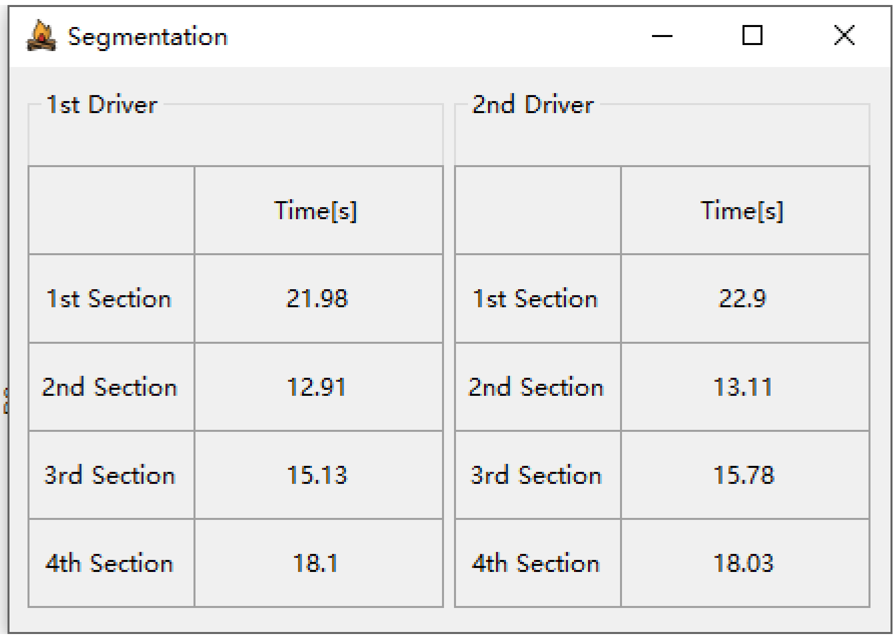
Related Functions:

```
void sensorChart_MouseMove(object sender, MouseEventArgs e)
private void sensorChart_MouseClick(object sender, MouseEventArgs e)
private void sensorChart_MouseDown(object sender, MouseEventArgs e)
private void sensorChart_MouseUp(object sender, MouseEventArgs e)
```



3.10 Segmentation

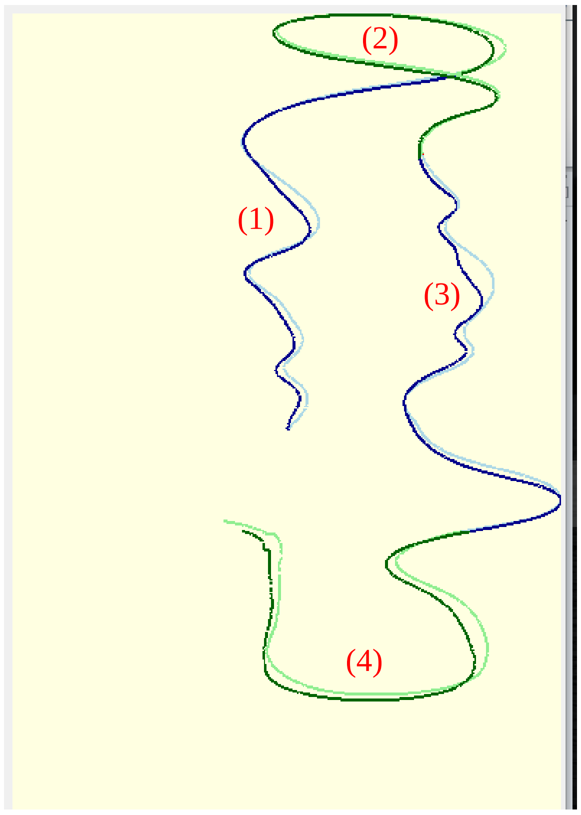
When we click on “Segmentation” button, there will be a window coming out, which shows the time it takes to complete every section for each driver. So we can compare the performance of the two drivers easily.



Segmentation

1st Driver		2nd Driver	
	Time[s]		Time[s]
1st Section	21.98	1st Section	22.9
2nd Section	12.91	2nd Section	13.11
3rd Section	15.13	3rd Section	15.78
4th Section	18.1	4th Section	18.03

On the map, it can show the four segmentations. The lighter color represents the first driver and the darker color represents another driver.



Related Functions:
private void segmentationButton_Click(object sender, EventArgs e)

4. Testing Environment

- Visual Studio 2013&2015
- Programming Language: C#