VTALogger Data Recording Application

Version 1.02a
December 2018

1. Introduction

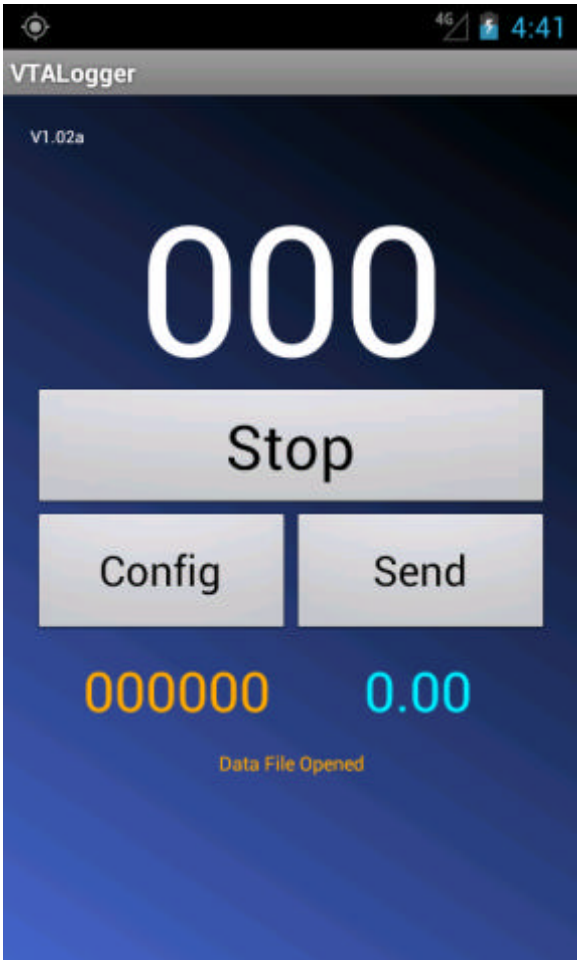
VTALogger is an Android mobile phone application for recording vehicle dynamic data. It should work successfully on any version of Android equal to or later than Version 4.1.2 but performs best on a phone running more recent versions of Android.

2. Installation

2.1. Software

To install the program you will need access to the storage on the phone. Typically you will be able to connect a computer to to the phone by using a USB cable. You can then access the phone from the computer as an external storage device.

You need to first create a directory named "RoadData" on the "Phone" storage device. A standard phone will have one or two storage options. The internal storage will be named "Phone" and any external memory card will be named "SDCard".



Copy the files VTALogger.Cfg and VTALogger.apk to this directory. The CFG file contains the initial identification and connection data for the program. The data in this file can be altered from within the VTALogger app.

You should then install the APK file. There can be a few different ways to do this. Some free third-party applications are available (such as ES File Explorer). More recent versions of Android may also provide the capability to install the APK file (for example the “MyFiles” app on Samsung phones).

The app should then be seen on the applications screen. You could then drag the app icon to the main display of the phone if you wish.

The app can be run by tapping the screen icon. After it has started you can select the CONFIG button and set the appropriate connection information for your situation. Most of the default settings can be kept, but you should set a unique user ID so your data can be identified from other data files. The recommended ID is a 4 character string where the first two characters indicate a text reference code and the second two are a number. For example, the ID may be AA01, where

"AA" indicates the user and "01" indicates a vehicle number related to that user.

2.2. Hardware

Mounting the phone in the vehicle requires a few things to be done. Ideally the phone should be mounted so it is vertical (portrait mode) and facing toward the driver. This is the arrangement assumed for the sensor alignment in our analysis programs.

The phone needs to be mounted firmly, but not rigidly to the vehicle. If it's mounted rigidly the accelerometers in the phone will pick up vibrations in the vehicle structure as well as the dynamic motion. If the phone is fixed too loosely the accelerometers will pick up the motion of the phone on its mount. A phone cradle with a robust structure works well. We've also had very good results using double-sided tape (Sellotape Mounting Tape - code 994004).

3. Operation

3.1. Start/Stop

Once the app has been installed you can start it by touching the app icon (a mid-blue circle). The app will open and display 3 buttons; **Start**, **Config** and **Send**.

To start recording just press the **Start** button. The button name will change to **Stop**. The app will start recording GPS and accelerometer data to a file on the phone's storage, in a directory named "RoadData".

When you have finished recording data select the **Stop** button. You can then record further data using the **Start** button, or send the recorded data to a web server using the **Send** button.

If you decide to send the data you should select the **Send** button. The data transfer can be a little slow if a lot of data has been recorded, so be patient. All compressed (Zip) data files in the "RoadData" directory will be sent to the defined location on the Internet. The original data files will be retained on the phone as a backup for a short period of time.

If the "RoadData" directory contains data files with both CSV and ZIP file extensions then that indicates the file data has not been sent to the web server. If the data has been sent then only ZIP files should remain.

The location where the data files are sent is defined in a **VTALogger.Cfg** configuration file. This file is created or altered using the **Config** option on the main app screen. It specifies the server location and also how long the CSV data files will be retained on the phone.

When the app is running various items of information are written to a file named **VTALogger.Txt**, located in the "RoadData" directory on the phone. This is a text log file which can be used to check some aspects of the program operation and verify files have been sent to the Internet.

3.2. Config Button

This button opens a configuration screen where you can set connection data and a vehicle or driver identification code. The program is able to send data files to a designated web location using FTP as the transfer standard. Please see section 5 of this document for more information.

This screen also has 5 buttons on it. There are two buttons to **Load** and **Save** the **VTALogger.Cfg** file. The **Done** button will close the **Config** screen and return to the main program screen. There is also an **About** button, to display some information about where to obtain further information about the application and operation. There is a **Calib** button to allow a short file to be recorded to help with later data calibration.

3.3. Send Button

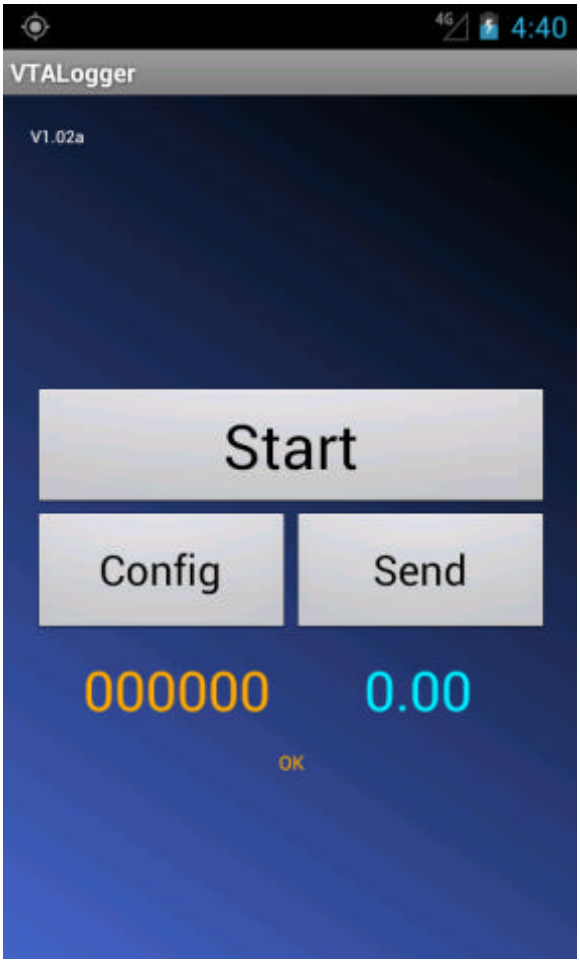
This will send all the data files (files with a VTA suffix) from the "RoadData" directory on the phone to the designated FTP site on the Internet. The FTP connection details are those set using the **Config** screen.

4. Screen Layout

The aim of the screen layout is to provide useful information without making the appearance and operation too complex. The screen offers three buttons and three sets of output data. The GPS (UTC) time is displayed in orange at the lower left of the screen. The distance travelled since the **Start** button was pressed is shown on the lower right in light blue. This is in units of kilometres. Finally the GPS velocity (in kilometres per hour) is shown in large white text in the top centre of the screen.

For the operation, there is a **Start** button to start recording vehicle data. This changes to a **Stop** button once recording starts. There is also a **Send** button, to send any available data files to a specified web location. The web location can be defined on a configuration menu, accessed using the **Config** button. Finally there is a **Calib** (calibration) button. See section 5.1 of this document for more information about this.

5. Configuration



You are able to set an FTP site where recorded data can be sent. This is defined using the **Config** screen. To do this you need to define the FTP address, user name and password. The FTP address will be of the form "ftp.website.com". When this site is first created a user name and password will have been set.

You should also set a unique driver/vehicle identifier (Driver ID). By default this is set to be "CC00". By defining a different ID it will be possible to differentiate between the data files from multiple vehicles. It is recommended you use a two letter code followed by a two digit number, such as "AB02".

Finally there is a parameter called "Days to Keep". This is a number representing the number of days the data files will be stored on the phone before they are deleted. This provides a degree of data backup. Typically you would record data then **SEND** it to the FTP site soon after it has been recorded. The data files can then be stored on the phone for a number of days. This allows you to get copies of the data files if any problems occurred with the original data transfer. You don't want the data files to occupy valuable phone storage for long periods of time, so they will be deleted after the defined number of days.

5.1. Calib Button

This button allows you to record a short calibration data file. It will record data for 100 seconds (GPS data records). The file created will have the file prefix "CAL" and the file suffix "VTA".

The typical use of the calibration data is to store a reference set of data for the stationary vehicle, before any formal testing commences. Normally when the phone app is first started there will be several seconds where the GPS establishes a positional fix. Once a fix is available the UTC time display on the app screen will start showing a time value. Ideally the vehicle should be stationary and level when a CAL file is recorded. This file will allow any accelerometer and orientation sensor misalignment to be calculated later.

6. Android Versions

Two versions of the APK file have been provided. An APK file is the installation package used to install appplications on an Android device. The file,

VTALogger.apk

is the standard installation package and will be suitable for any version of Android later than 4.1.2. As it is sometimes appropriate to use an older device for data recording we also provide the file,

VTALogger_V230.apk

This is able to be run on earlier versions of Android. It is intended for Android versions 2.3.1 and later. Both apps will appear the same on the device and function in the same way.

7. Orientation Sensor Values

We currently record sensor data from the Android device using the Android orientation sensor. This was deprecated from Android version 2.2 (API Level 8) so the results can be in error depending on how you're recording and have your device oriented. In general the orientation values should be consistent provided there is no roll angle present. We will endeavor to improve the calculation of orientation angles once other parts of the software have been addressed.

8. Format

The data files produced by the **VTALogger** program have a consistent structure. Although the file suffix is VTA the file itself is a Comma-Separated Variable (CSV) text file, so can be imported into many spreadsheet programs or easily viewed with a simple text editor.

The file name format is typically as follows.

VT15082016_074940_CC02.Vta

where there is a two character prefix followed by a date and time value. These are for the date and time the data file was first created. This is then folowed by the driver/vehicle ID.

The internal file structure contains 3 lines of header data, as follows.

%% VTALogger Version: 1.02a
$UTCDate,UTCTime,Latitude,Longitude,Altitude,Speed,Bearing,NumSat
#Idx,Time,Event,OX,OY,OZ,GX,GY,GZ

then the bulk of the recorded data is found, with the file terminated with a **%% End** string.

8.1. Timing

The GPS records at one hertz (updates per second). The IMU records at 80 to 100 hertz (nominally). The GPS time reference comes from an atomic clock so it should be the reference for all timing. The IMU output includes an elapsed time value relative to when recording commence on the phone. Reference to external events should always be based on the GPS UTC time value.

8.2. Format Examples

GPS	$81116,5435800,-37896827,145042923,3.27,7,10849,9
IMU	#1210.69,0,-1.4,3.9,3.5,-0.020,0.068,0.997

8.3. GPS Data Record

Each line starts with the "$" character. There are 8 data items on each line. These are,

Date		(Day-Month-Year)
UTC Time		(Hour-Minute-Second-Millisecond)
Latitude		(degrees)
Longitude		(degrees)
Altitude		(metres)
Velocity		(kmh)
Bearing		(degrees)
No of Satellites	(integer)	must be at least 4 for a 3D fix

8.4. IMU Data Record

Each line starts with the "#" character. There are 8 data items on each line. These are,

Elapsed Time	(seconds)	from micro-processor start
Event Number (integer)	starting at 0 (not used yet)	
Roll Angle	(degrees)	
Pitch Angle	(degrees)	
Yaw Angle	(degrees)	compass heading, from 0 to 360
GX	(m/sec^2)	in the direction of travel
GY	(m/sec^2)	laterally
GZ	(m/sec^2)	vertically

9. Contact

For information about this application you can use this e-mail address.

info@temporal.com.au

The **VTALogger** program is free to use and distribute. If you wish to obtain an analysis program to display the recorded data we will be offering several programs to do this. These analysis programs will be available from this web link.

www.testcell3.com

Appendix 1: Sensor Alignment

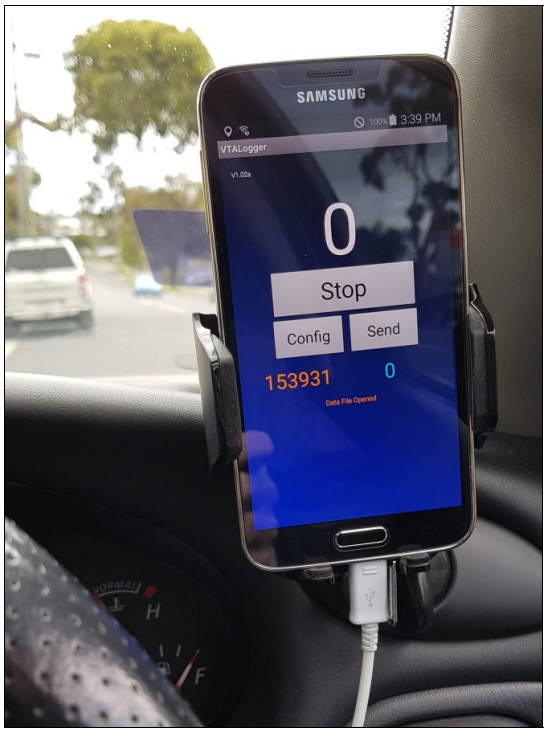
When looking at the phone screen the axes of the phone will have X to right of phone screen, Y pointing to the top of the phone and Z pointing out of the screen toward you.

The orientation sensor derives its data by using a device's geomagnetic field sensor in combination with a device's accelerometer. Using these two hardware sensors, an orientation sensor provides data for the following three dimensions:

Azimuth (degrees of rotation around the Z axis). This is the angle between magnetic north and the device's Y axis. For example, if the device's Y axis is aligned with magnetic north this value is 0, and if the device's Y axis is pointing south this value is 180. Likewise, when the Y axis is pointing east this value is 90 and when it is pointing west this value is 270.

Pitch (degrees of rotation around the X axis). This value is positive when the positive Z axis rotates toward the positive Y axis, and it is negative when the positive Z axis rotates toward the negative Y axis. The range of values is 180 degrees to -180 degrees.

Roll (degrees of rotation around the Y axis). This value is positive when the positive Z axis rotates toward the positive X axis, and it is negative when the positive Z axis rotates toward the negative X axis. The range of values is 90 degrees to -90 degrees.

This definition is different from yaw, pitch, and roll used in aviation, where the X axis is along the longitudinal axis of an aircraft (tail to nose). Also, for historical reasons the roll angle is positive in the clockwise direction (mathematically speaking, it should be positive in the counter-clockwise direction).

The orientation sensor derives its data by processing the raw sensor data from the accelerometer and the geomagnetic field sensor. Because of the heavy processing that is involved, the accuracy and precision of the orientation sensor is diminished (specifically, this sensor is only reliable when the roll component is 0).