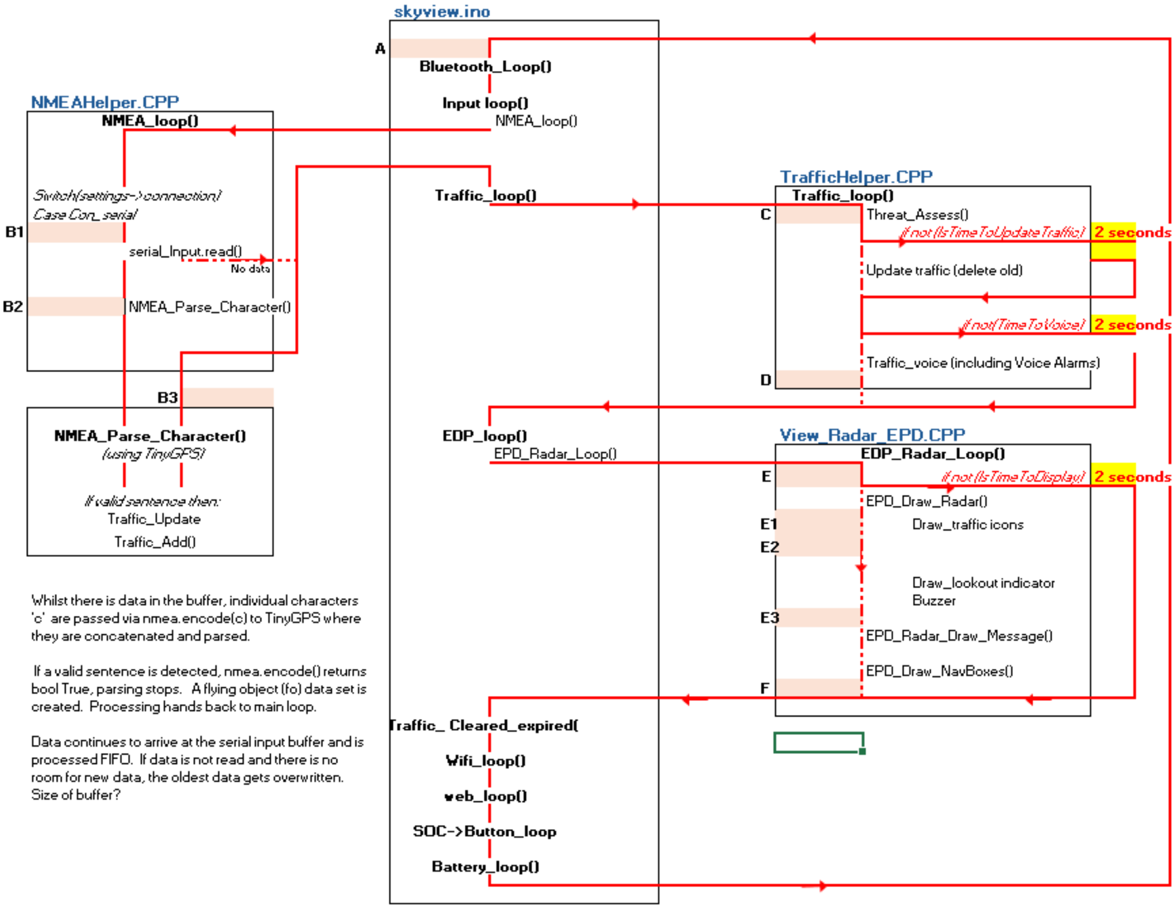
SkyView – Loop Timing Analysis Tools and Method

Timing markers and loop diagram

Insert timing marker Serial.println lines into the sketch. These will be output via the USB serial port like debug printing. Eg Serial.println (“Timer A”). Note the ‘ln’ adds a line feed after the print.

 Use loop diagram to understand where to place timing markers. (Loop diagram is available as separate doc).



Whilst there is data in the buffer, individual characters 'c' are passed via nmea.encode(c) to TinyGPS where they are concatenated and parsed.

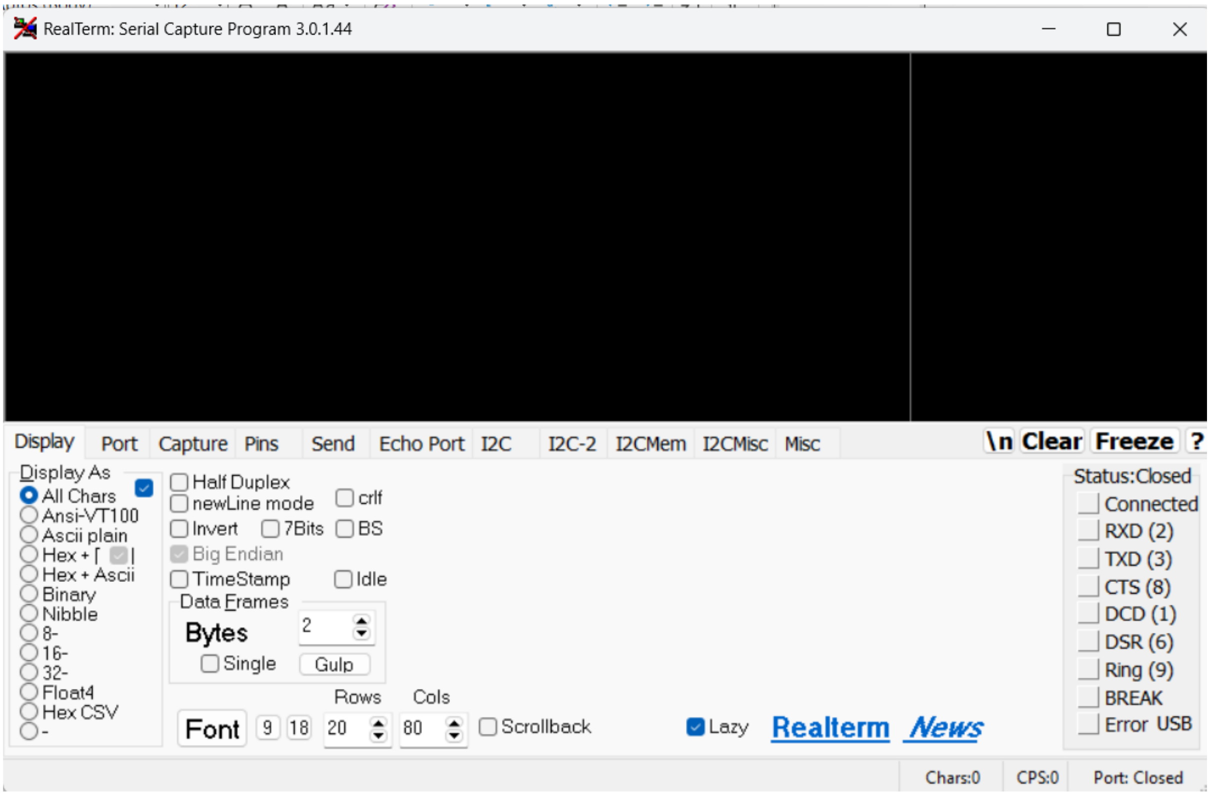
If a valid sentence is detected, nmea.encode() returns bool True, parsing stops. A flying object (fo) data set is created. Processing hands back to main loop.

Data continues to arrive at the serial input buffer and is processed FIFO. If data is not read and there is no room for new data, the oldest data gets overwritten. Size of buffer?

Down load RealTerm serial monitor and capture program

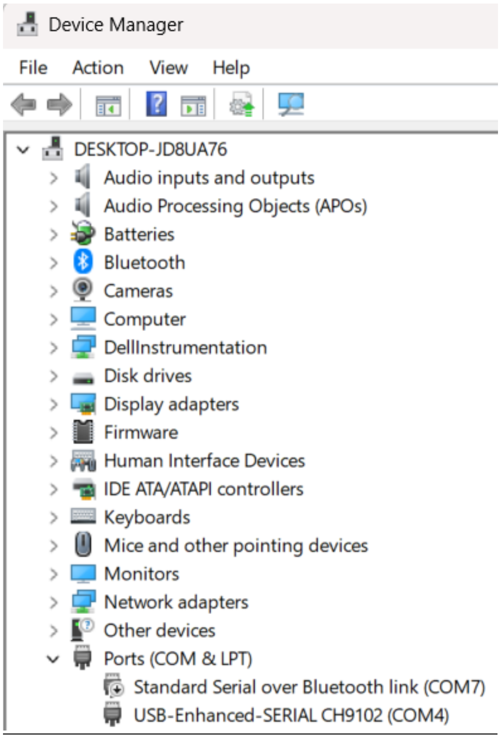
RealTerm is a free download from the Web. The advantage of RealTerm is that it can capture data with timestamps in milli seconds (ms).

It’s not obvious when you first use the program how to get the timestamps in ms, so follow the instructions below.



Select port and data format

Use device manager to discover port number. In this example Port 4. You may need to find and load the driver.

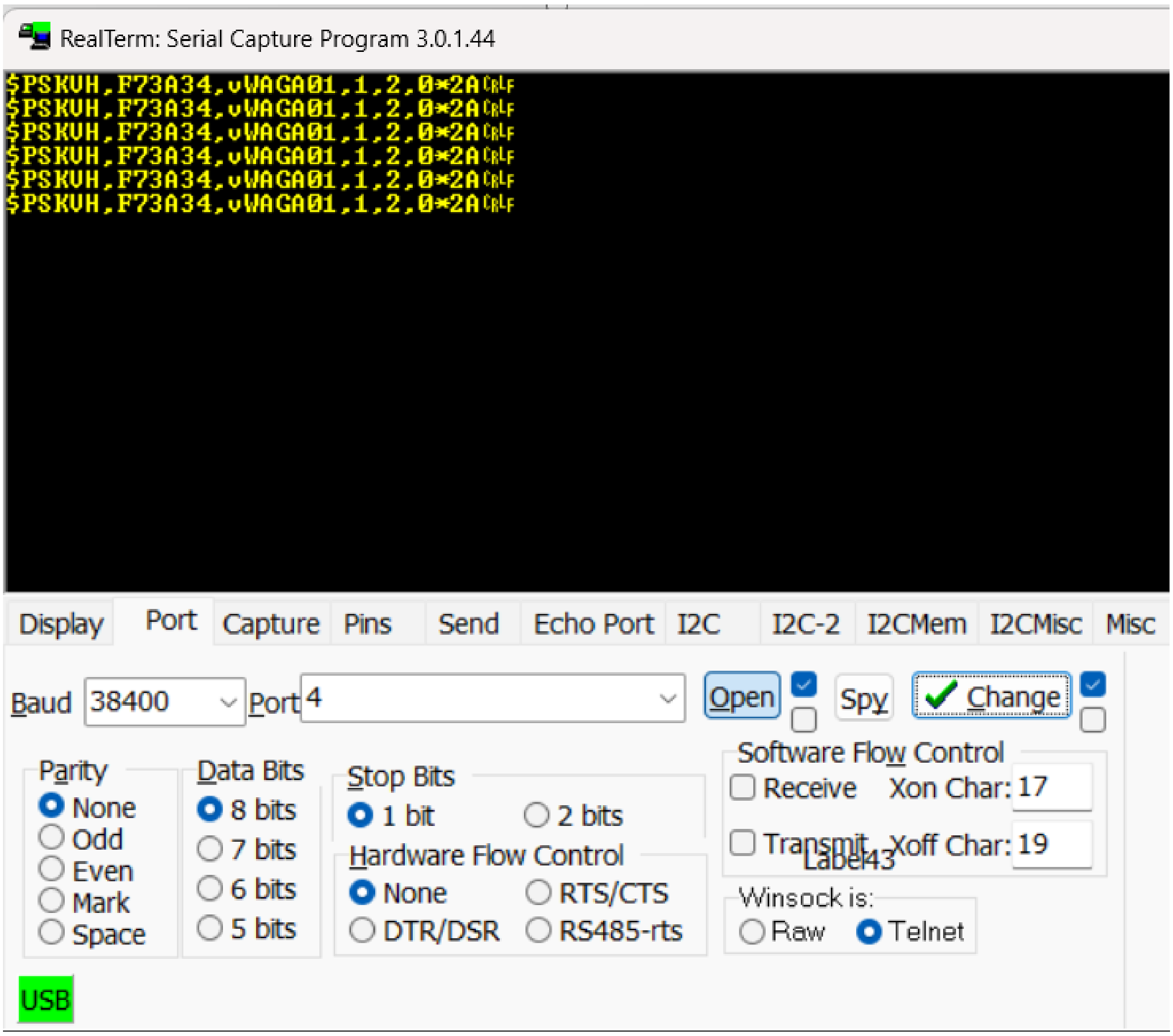


Device Manager

DESKTOP-JD8UA76
- Audio inputs and outputs
- Audio Processing Objects (APOs)
- Batteries
- Bluetooth
- Cameras
- Computer
- DellInstrumentation
- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Ports (COM & LPT)
  - Standard Serial over Bluetooth link (COM7)
  - USB-Enhanced-SERIAL CH9102 (COM4)

In RealTerm, go to the Port tab. Select the required Port number and the standard SkyView Baud rate 38400. Click ‘Change’.

Check Parity= None, Data Bits=8 bits, Stop Bits=1 bit. This is the standard 8N1 setting.



RealTerm: Serial Capture Program 3.0.1.44

```
$PSKVH,F73A34,vWAGA01,1,2,0*2ACRLF
$PSKVH,F73A34,vWAGA01,1,2,0*2ACRLF
$PSKVH,F73A34,vWAGA01,1,2,0*2ACRLF
$PSKVH,F73A34,vWAGA01,1,2,0*2ACRLF
$PSKVH,F73A34,vWAGA01,1,2,0*2ACRLF
$PSKVH,F73A34,vWAGA01,1,2,0*2ACRLF
```

Baud 38400 Port 4

Parity: ● None ○ Odd ○ Even ○ Mark ○ Space

Data Bits: ● 8 bits ○ 7 bits ○ 6 bits ○ 5 bits

Stop Bits: ● 1 bit ○ 2 bits

Hardware Flow Control: ● None ○ RTS/CTS ○ DTR/DSR ○ RS485-rts

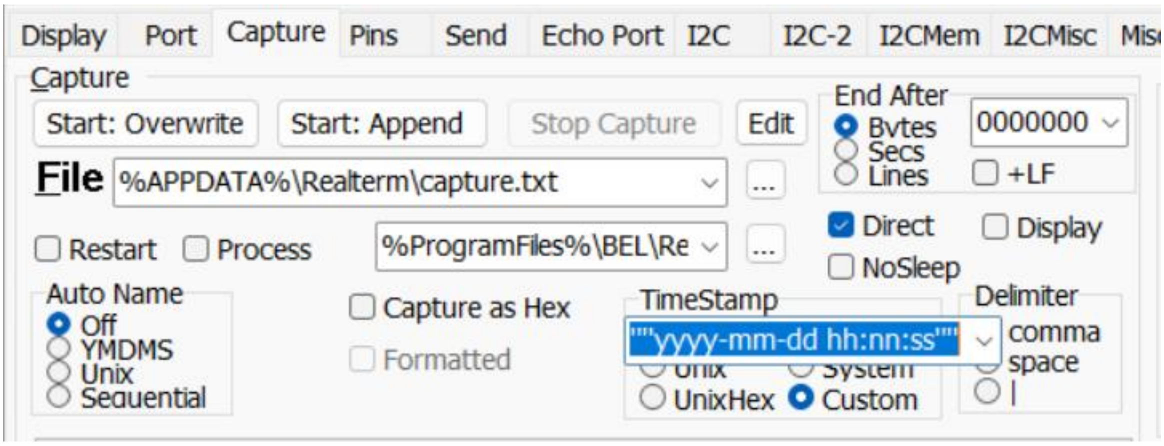
Software Flow Control: □ Receive Xon Char: 17 □ Transmit Xoff Char: 19

Winsock is: ○ Raw ● Telnet

In the above screenshot, a SkyView is issuing its $PSKVH ‘pulse’ which shows its ID , version number. Note the 2 figure checksum after the * (ie *2A). Each line ends with CRLF (Carriage return and Line Feed).

Change timestamp format to include milli seconds

Select the Capture Tab. Set TimeStamp =Custom. Right click anywhere in the TimeStamp box to reveal the timestamp format.



Capture

File %APPDATA%\Realterm\capture.txt

□ Restart □ Process %ProgramFiles%\BEL\Re

Auto Name: ● Off ○ YMDMS ○ Unix ○ Sequential

□ Capture as Hex □ Formatted

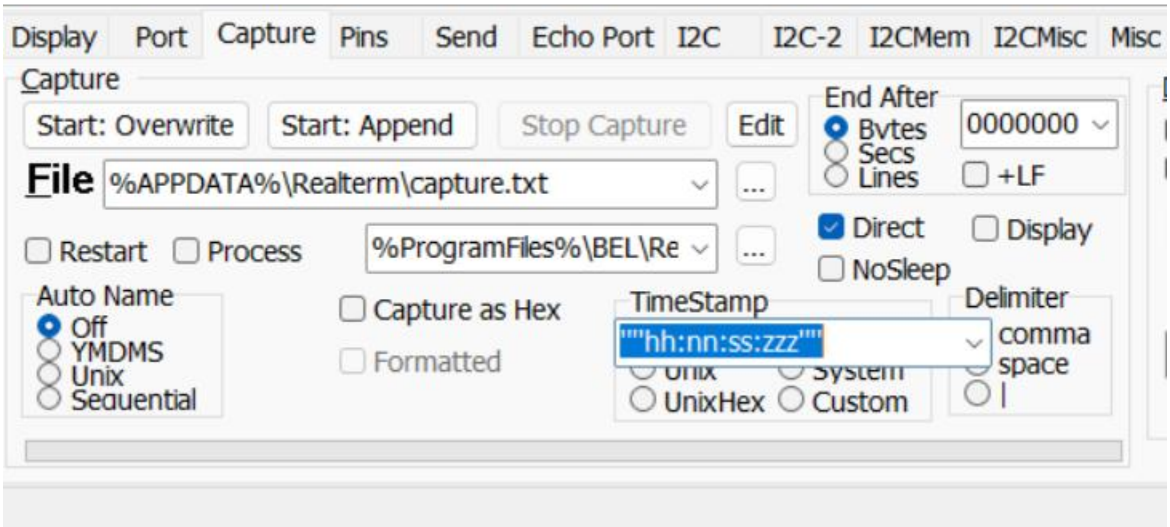
End After: ● Bytes ○ Secs ○ Lines 0000000 □ +LF

☑ Direct □ Display □ NoSleep

TimeStamp: ""yyyy-mm-dd hh:nn:ss"" ○ Unix ○ System ○ UnixHex ● Custom

Delimiter: comma space ○ |

Edit the timestamp format to include milli seconds :zzz. Suggest delete yyyy-mm-dd.



Capture

File %APPDATA%\Realterm\capture.txt

□ Restart □ Process %ProgramFiles%\BEL\Re

Auto Name: ● Off ○ YMDMS ○ Unix ○ Sequential

□ Capture as Hex □ Formatted

End After: ● Bytes ○ Secs ○ Lines 0000000 □ +LF

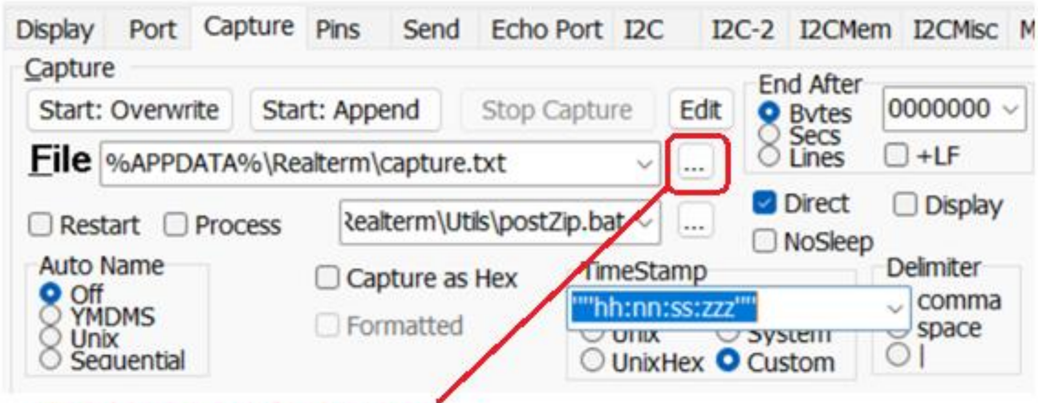
☑ Direct □ Display □ NoSleep

TimeStamp: ""hh:nn:ss:zzz"" ○ Unix ○ System ○ UnixHex ○ Custom

Delimiter: comma space ○ |

Setup file for capture

Select/set file name for capture output using file browser.



Capture

File %APPDATA%\Realterm\capture.txt

□ Restart □ Process Realterm\Utils\postZip.bat

Auto Name: ● Off ○ YMDMS ○ Unix ○ Sequential

□ Capture as Hex □ Formatted

End After: ● Bytes ○ Secs ○ Lines 0000000 □ +LF

☑ Direct □ Display □ NoSleep

TimeStamp: ""hh:nn:ss:zzz"" ○ Unix ○ System ○ UnixHex ● Custom

Delimiter: comma space ○ |

Click here to get file browser

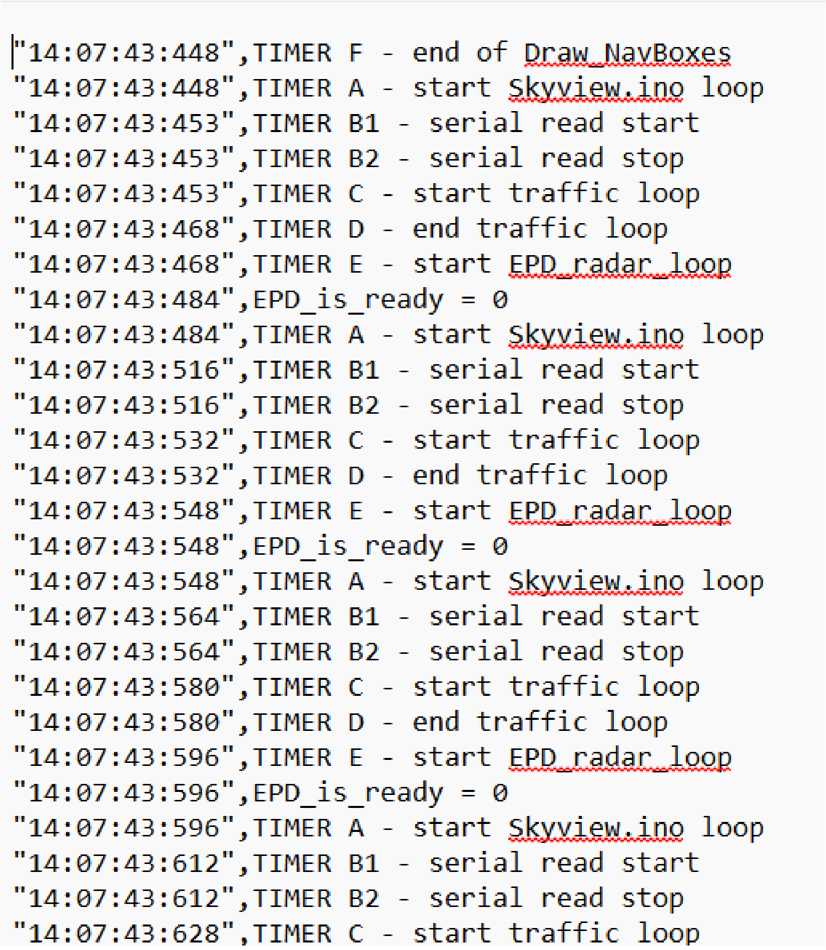
Set 'End After' if you wish or else let it run until you stop it.

Click 'Start:Overwrite' or' Start:Append' to output capture file.

RealTerm screen will be blank as the capture is in progress.

RealTerm capture output

The capture file will be a .txt text file. See example below displayed with notepad.

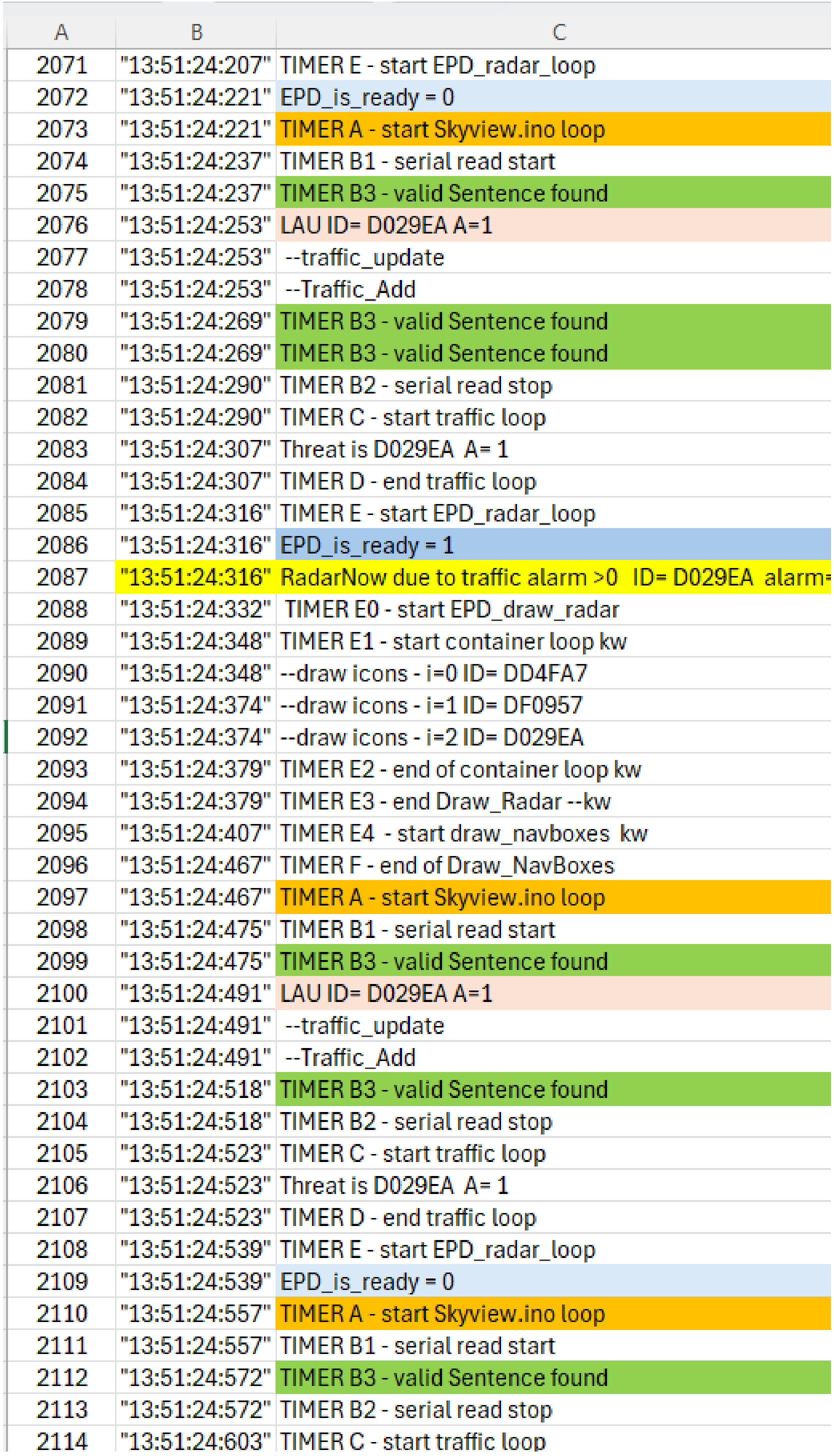


```
"14:07:43:448",TIMER F - end of Draw_NavBoxes
"14:07:43:448",TIMER A - start Skyview.ino loop
"14:07:43:453",TIMER B1 - serial read start
"14:07:43:453",TIMER B2 - serial read stop
"14:07:43:453",TIMER C - start traffic loop
"14:07:43:468",TIMER D - end traffic loop
"14:07:43:468",TIMER E - start EPD_radar_loop
"14:07:43:484",EPD_is_ready = 0
"14:07:43:484",TIMER A - start Skyview.ino loop
"14:07:43:516",TIMER B1 - serial read start
"14:07:43:516",TIMER B2 - serial read stop
"14:07:43:532",TIMER C - start traffic loop
"14:07:43:532",TIMER D - end traffic loop
"14:07:43:548",TIMER E - start EPD_radar_loop
"14:07:43:548",EPD_is_ready = 0
"14:07:43:548",TIMER A - start Skyview.ino loop
"14:07:43:564",TIMER B1 - serial read start
"14:07:43:564",TIMER B2 - serial read stop
"14:07:43:580",TIMER C - start traffic loop
"14:07:43:580",TIMER D - end traffic loop
"14:07:43:596",TIMER E - start EPD_radar_loop
"14:07:43:596",EPD_is_ready = 0
"14:07:43:596",TIMER A - start Skyview.ino loop
"14:07:43:612",TIMER B1 - serial read start
"14:07:43:612",TIMER B2 - serial read stop
"14:07:43:628",TIMER C - start traffic loop
```

Import to Excel

Import as a CSV file to split time and info into different columns.

Create a sequential number in column A. Sort by Timer/Info description and apply colours to aid analysis. Re-sort using column A to restore original order.



A	B	C
2071	"13:51:24:207"	TIMER E - start EPD_radar_loop
2072	"13:51:24:221"	EPD_is_ready = 0
2073	"13:51:24:221"	TIMER A - start Skyview.ino loop
2074	"13:51:24:237"	TIMER B1 - serial read start
2075	"13:51:24:237"	TIMER B3 - valid Sentence found
2076	"13:51:24:253"	LAU ID= D029EA A=1
2077	"13:51:24:253"	--traffic_update
2078	"13:51:24:253"	--Traffic_Add
2079	"13:51:24:269"	TIMER B3 - valid Sentence found
2080	"13:51:24:269"	TIMER B3 - valid Sentence found
2081	"13:51:24:290"	TIMER B2 - serial read stop
2082	"13:51:24:290"	TIMER C - start traffic loop
2083	"13:51:24:307"	Threat is D029EA A= 1
2084	"13:51:24:307"	TIMER D - end traffic loop
2085	"13:51:24:316"	TIMER E - start EPD_radar_loop
2086	"13:51:24:316"	EPD_is_ready = 1
2087	"13:51:24:316"	RadarNow due to traffic alarm >0 ID= D029EA alarm=
2088	"13:51:24:332"	TIMER E0 - start EPD_draw_radar
2089	"13:51:24:348"	TIMER E1 - start container loop kw
2090	"13:51:24:348"	--draw icons - i=0 ID= DD4FA7
2091	"13:51:24:374"	--draw icons - i=1 ID= DF0957
2092	"13:51:24:374"	--draw icons - i=2 ID= D029EA
2093	"13:51:24:379"	TIMER E2 - end of container loop kw
2094	"13:51:24:379"	TIMER E3 - end Draw_Radar --kw
2095	"13:51:24:407"	TIMER E4 - start draw_navboxes kw
2096	"13:51:24:467"	TIMER F - end of Draw_NavBoxes
2097	"13:51:24:467"	TIMER A - start Skyview.ino loop
2098	"13:51:24:475"	TIMER B1 - serial read start
2099	"13:51:24:475"	TIMER B3 - valid Sentence found
2100	"13:51:24:491"	LAU ID= D029EA A=1
2101	"13:51:24:491"	--traffic_update
2102	"13:51:24:491"	--Traffic_Add
2103	"13:51:24:518"	TIMER B3 - valid Sentence found
2104	"13:51:24:518"	TIMER B2 - serial read stop
2105	"13:51:24:523"	TIMER C - start traffic loop
2106	"13:51:24:523"	Threat is D029EA A= 1
2107	"13:51:24:523"	TIMER D - end traffic loop
2108	"13:51:24:539"	TIMER E - start EPD_radar_loop
2109	"13:51:24:539"	EPD_is_ready = 0
2110	"13:51:24:557"	TIMER A - start Skyview.ino loop
2111	"13:51:24:557"	TIMER B1 - serial read start
2112	"13:51:24:572"	TIMER B3 - valid Sentence found
2113	"13:51:24:572"	TIMER B2 - serial read stop
2114	"13:51:24:603"	TIMER C - start traffic loop

Analysis

As you wish!

Timp: To convert the time stamp to seconds use excel formula (where B2105 is the cell to be evaluated):

=VALUE(MID(B2105,8,2))+VALUE(MID(B2105,11,3))/1000

= 24.523 seconds