Some/IP Service Implementation

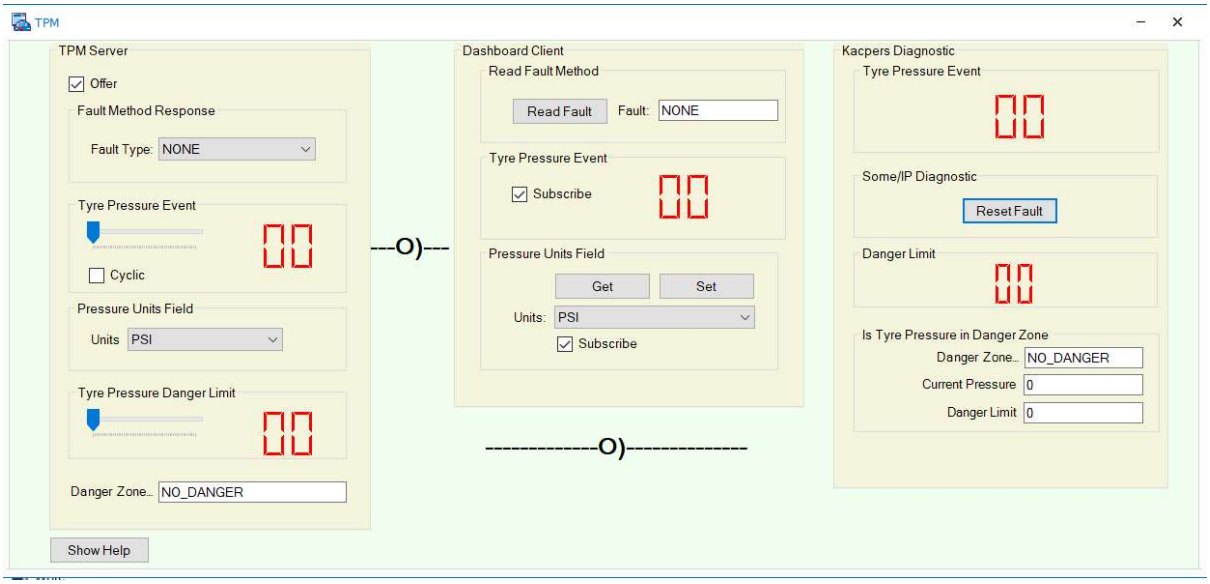
By Kacper Woloszyn- 20071494

GITHUB LINK AND DOCUMENTATION: https://github.com/kacper97/SomeIp_ConnectedCar

Contents

Introduction.

In this report I will put in the things that I've included. The specifications of the assignment were followed, and all the steps were implemented. I have the ids of all the things I've added specified in the variables part of both server and sub2



GUI DIAGRAM

Specifications:

- ## Server

- `const DWORD DANGER_PRESSURE_EVENT = 0x8003;  // danger pressure event has id 0x8003`
- `const DWORD ZONE_EVENT = 0x8005;  // zone event has id 0x8005`
- `const DWORD ResetSetterID = 33;  // reset setter  id`
- `const DWORD ResetGetterId = 34;  // reset getter id`
- `const DWORD DanGetterId =35;  //danger limit getter id`
- `const DWORD setDangerLimitlowID = 37;  // setter low danger limit id`
- `const DWORD getDangerLimitLowID = 38;  // getter low danger limit id`
- `const DWORD setDangerLimitHighID =40; // setter high danger limit id`
- `const DWORD getDangerLimitHighID =41;  // getter high danger limit id`
- `const DWORD DanSetterId =36;  // danger limit setter id`
- `const DWORD danZoneGetterID =45;  // danger zone getter id`

- ## SUB2

- `const DWORD TYRE_PRESSURE_EVENT = 0x8001;  //Tyre pressure Id`
- `const DWORD DANGER_PRESSURE_EVENT = 0x8003;  // Danger limit id`
- `const DWORD ZONE_EVENT = 0x8005;  // Danger zone event`
- `const DWORD FieldGetterID = 31;  // Get Pressure id`
- `const DWORD ResetSetterID = 33;  // Reset Id`
- `const DWORD setDangerLimitlowID = 37; // 40 danger limit id`
- `const DWORD setDangerLimithighID = 40; // 50 danger limit id`

SERVER:

Variables:

- `DWORD danger_limit;  // Event handle for tyre danger limit`
- `DWORD reset_b;  // Field Handle for Reset Error`
- `DWORD dan_zone_limit; // Event handle for danger zone limit`
- `DWORD danlimit_low;  // Field Handle for setting limit to 40`
- `DWORD danlimit_high;  // field handle for setting limit to 50`
- `DWORD dan_limit_field;  // field handle for danger`
- `const DWORD DANGER_PRESSURE_EVENT = 0x8003;  // danger pressure event has id 0x8003`
- `const DWORD ZONE_EVENT = 0x8005;  // zone event has id 0x8005`
- `const DWORD ResetSetterID = 33;  // reset setter  id`
- `const DWORD ResetGetterId = 34;  // reset getter id`
- `const DWORD DanGetterId =35;  //danger limit getter id`
- `const DWORD setDangerLimitlowID = 37;  // setter low danger limit id`
- `const DWORD getDangerLimitLowID = 38;  // getter low danger limit id`
- `const DWORD setDangerLimitHighID =40; // setter high danger limit id`
- `const DWORD getDangerLimitHighID =41;  // getter high danger limit id`
- `const DWORD DanSetterId =36;  // danger limit setter id`
- `const DWORD danZoneGetterID =45;  // danger zone getter id`

Void StartServer:

```
//danger
danger_limit = SomeIpAddEvent(psi, DANGER_PRESSURE_EVENT,"Send_Dan_Limit");
CheckForSomeIpError();

//danger zone
dan_zone_limit = SomeIpAddEvent(psi, ZONE_EVENT, "Send_dan_zone_limit");
CheckForSomeIpError();

// adding limit to event group
SomeIpAddEventToEventgroup(peg, danger_limit);
CheckForSomeIpError();

// adding zone limit to event group
SomeIpAddEventToEventgroup(peg, dan_zone_limit);
CheckForSomeIpError();

// create another Eventgroup for the field notification
peg2 = SomeIpAddProvidedEventGroup(psi,EG2 );
CheckForSomeIpError();

//G
SomeIpAddMethod(psi,12,"onDangerZoneRequest");
CheckForSomeIpError();

// Initialise the fields to the default pressure units,Fault Type, And
Danger Zone according to variable TPMS::Units
dataBuffer[0] = @TPMS::Units;;
dataBuffer[0] = @TPMS::FaultType;
dataBuffer[0] = @TPMS::DangerZone;

//Reset
SomeIpAddMethod(psi,33,"OnSomeIpResetFieldRequest");
CheckForSomeIpError();
//Danger Limit
SomeIpAddMethod(psi,34,"Dan_Limit_Field_Request");
CheckForSomeIpError();
//40
SomeIpAddMethod(psi,37,"OnFortySet");
CheckForSomeIpError();
//50
SomeIpAddMethod(psi,40,"OnFiftySet");
CheckForSomeIpError();
```

Method to get the current danger limit

```
/*********************************************************************
*  Handler for DangerLimit() method.
*  Sets the field value to desired value and also update the system
variable
*  TPMS::DangerLimit
*********************************************************************/
void Dan_Limit_Field_Request( dword methodHandle, dword messageHandle,
dword messageResponseHandle )
{
byte dataBuffer[1];

// Retrieve the method parameter from SOME/IP method call.
SomeIpGetData(messageHandle,1,dataBuffer);
CheckForSomeIpError();
@TPMS::DangerLimit = dataBuffer[0];
SomeIpSetData(dan_limit_field,1,dataBuffer);
CheckForSomeIpError();

// commit field content ... notification is sent
SomeIpCommitField(dan_limit_field);
CheckForSomeIpError();

// Echo the field value in the Setter response
SomeIpSetData (messageResponseHandle,1,dataBuffer);
CheckForSomeIpError();
}
```

Method to Reset the Fault

```
/*********************************************************************
*  Method that resets the fault type to be None when the user presses R or
the button
*********************************************************************/
void OnSomeIpResetFieldRequest( dword methodHandle, dword messageHandle,
dword messageResponseHandle )
{

byte dataBuffer[1];

// Retrieve the meth  od parameter from SOME/IP method call.
SomeIpGetData(messageHandle,1,dataBuffer);
CheckForSomeIpError();

@TPMS::FaultType = 0;
// Echo the field value in the Setter response
SomeIpSetData (messageResponseHandle,1,dataBuffer);
CheckForSomeIpError();
}
```

Method when 4 is pressed to set danger limit to 40

```
/*********************************************************************
*  Method that sets the danger limit value to be 40 when the user presses 4
*********************************************************************/
void OnFortySet( dword methodHandle, dword messageHandle, dword
messageResponseHandle )
{
byte dataBuffer[1];

// Retrieve the method  od parameter from SOME/IP method call.
SomeIpGetData(messageHandle,1,dataBuffer);
CheckForSomeIpError();

@TPMS::DangerLimit = 40;
// Echo the field value in the Setter response

SomeIpSetData (messageResponseHandle,1,dataBuffer);
CheckForSomeIpError();
}
```

Method when 5 is pressed to set danger limit to 50

```
/*********************************************************************
*  Method that sets the danger limit value to be 50 when the user presses 5
*********************************************************************/
void OnFiftySet( dword methodHandle, dword messageHandle, dword
messageResponseHandle )
{

byte dataBuffer[1];

// Retrieve the method  od parameter from SOME/IP method call.
SomeIpGetData(messageHandle,1,dataBuffer);
CheckForSomeIpError();
@TPMS::DangerLimit = 50;

// Echo the field value in the Setter response

SomeIpSetData (messageResponseHandle,1,dataBuffer);
CheckForSomeIpError();
}
```

Method that sends the tyre pressure and gets the danger limit and sends the danger zone value if pressure is bigger than limit

```
/*********************************************************************
*  Send the tyrePressure event whenever the TPMS::Units variable changes.
*  Linking the variable to a panel slider control causes the new value to
*  be sent whenever the slider changes.
*  if the pressure is higher or
*  lower it sets the Dan_Zone_limit and triggers event
*********************************************************************/

on sysvar sysvar::TPMS::TyrePressure
{
   SomeIpTriggerEvent(pev);
   CheckForSomeIpError();

  //DangerZone
  if(@TPMS::TyrePressure>@TPMS::DangerLimit )
  {
  @TPMS::DangerZone=1;
   SomeIpTriggerEvent(dan_zone_limit);
   CheckForSomeIpError();
  }
  if(@TPMS::TyrePressure<@TPMS::DangerLimit)
  {
   @TPMS::DangerZone=0;
   SomeIpTriggerEvent(dan_zone_limit);
   CheckForSomeIpError();
  }
}
```

Method that sends the danger limit and gets the pressure value and sends the danger zone value if pressure is bigger than limit

```
/*********************************************************************
*  Sends the current danger limit value and if the pressure is higher or
*  lower it sets the Dan_Zone_limit and triggers event
*********************************************************************/
on sysvar sysvar::TPMS::DangerLimit
{
   SomeIpTriggerEvent(danger_limit);
   CheckForSomeIpError();
    // set value of field data

  //DANGER ZONE
  if(@TPMS::TyrePressure>@TPMS::DangerLimit )
  {
   @TPMS::DangerZone=1;
   SomeIpTriggerEvent(dan_zone_limit);
   CheckForSomeIpError();
  }
  if(@TPMS::TyrePressure<@TPMS::DangerLimit)
  {
   @TPMS::DangerZone=0;
   SomeIpTriggerEvent(dan_zone_limit);
   CheckForSomeIpError();
  }
}
```

Method to get the current danger zone value when G is pressed

```
/*********************************************************************
* Danger Slider, returns the value from the slider for the G button
*********************************************************************/
void onDangerZoneRequest( dword methodHandle, dword messageHandle, dword
messageResponseHandle )
{
  byte dataBuffer[1];
  dataBuffer[0] = @TPMS::DangerLimit;

  // Return the selected fault type from the TPMS panel.
  SomeIpSetData (messageResponseHandle,1,dataBuffer);
  CheckForSomeIpError();
}
```

Method that sends the current value of danger limit to the Sub2

```
/*********************************************************************
* Function that sends the current danger limit value to the Sub2.
*********************************************************************/
void Send_Dan_Limit(DWORD eventHandle, DWORD messageHandle)
{
  byte dataBuffer[1];

  writeEx(0,1,"\nTPMS-SOME/IP: Preparing to send danger limi event." );

  // Insert danger limit value in the event message.

  dataBuffer[0]=@TPMS::DangerLimit;
  SomeIpSetData(messageHandle,1,dataBuffer);
  CheckForSomeIpError();
}
```

Method that sends the current danger limit zone to Sub2

Sends either a 0 or 1 as in the table.

```
/*********************************************************************
* Function that sends the current danger limit zone value to the Sub2.
*********************************************************************/
void Send_dan_zone_limit(DWORD eventHandle, DWORD messageHandle)
{
  byte dataBuffer[1];

  writeEx(0,1,"\nTPMS-SOME/IP: Preparing to send danger zone value event."
);

  // Insert danger limit value in the event message.
  dataBuffer[0]=@TPMS::DangerZone;

  SomeIpSetData(messageHandle,1,dataBuffer);
  CheckForSomeIpError();
}
```

SUB2:

Variables:

```
DWORD dan_zone_limit;     // event call to get limit val
DWORD resetFaultCodeSetter;   //method call to reset the fault
DWORD setDangerLimitLow;      // method call to set 40 danger limit
DWORD setDangerLimitHigh;     // method call to set 50 danger limit
DWORD Current_dan_val_method;  // global to get g val
DWORD dan_limit;          // danger limit
const DWORD TYRE_PRESSURE_EVENT = 0x8001;  //Tyre pressure Id
const DWORD DANGER_PRESSURE_EVENT = 0x8003;  // Danger limit id
const DWORD ZONE_EVENT = 0x8005;        // Danger zone event
const DWORD FieldGetterID = 31;         // Get Pressure id
const DWORD ResetSetterID = 33;         // Reset Id
const DWORD setDangerLimitlowID = 37; // 40 danger limit id
const DWORD setDangerLimithighID = 40; // 50 danger limit id
```

On start

```
//Danger Limit Event
dan_limit = SomeIpCreateEventConsumer(csi,0x8003,"getDangerLimitMethod");
CheckForSomeIpError();

//Danger Zone Event
dan_zone_limit = SomeIpCreateEventConsumer(csi,0x8005,"OnZoneResponse");
CheckForSomeIpError();


//Pressure Event
gMcGetter =
SomeIpCreateMethodCall(csi,FieldGetterID,"OnFieldGetterResponse");
CheckForSomeIpError();

//RESET
resetFaultCodeSetter =
SomeIpCreateMethodCall(csi,ResetSetterID,"onFaultResetResponse");
CheckForSomeIpError();

//40
setDangerLimitLow =
SomeIpCreateMethodCall(csi,setDangerLimitlowID,"onDanFortyResponse");
CheckForSomeIpError();

//50
setDangerLimitHigh =
SomeIpCreateMethodCall(csi,setDangerLimithighID,"onDanFiftyResponse");
CheckForSomeIpError();

// G press
Current_dan_val_method = SomeIpCreateMethodCall(csi,12,"OnDangerResponse");
CheckForSomeIpError();

//Works also without any Fibex/ARXML but only with SomeIpSetData and not
SomeIpSetValueDWord (online help)
SomeIpSetData(Current_dan_val_method,elCount(payload),payload);
CheckForSomeIpError();
```

Method to get the current value of Danger Zone from server

```
/*********************************************************************
* Gets the current value of the Danger zone event from the server
*********************************************************************/
void OnZoneResponse(DWORD eventHandle, DWORD messageHandle)
{
  DWORD res;         // value of return parameter
  char data[1];

  // get the returned parameter values
  res = SomeIpGetData(messageHandle,elCount(data), data);
  CheckForSomeIpError();

  writeEx(0,1,"\nDASH-SOME/IP: Danger zone() method return value:  %d
",data[0]);

  @SUB2::DangerZone= data[0];
}
```

Methods that Write out to the write window in the sub2.

```
/*********************************************************************
* Method that writes out the Fault response
*********************************************************************/
void onFaultResetResponse(dword methodCallHandle, dword
messageResponseHandle )
{
writeEx(0,1,"\nSUB2-SOME/IP: Reset Fault Code() method response
received.");
}

/*********************************************************************
* Method to write out when 4 is pressed
*********************************************************************/
void onDanFortyResponse(dword methodCallHandle, dword messageResponseHandle
)
{
writeEx(0,1,"\nSUB2-SOME/IP: dan_limit 40 Code() method response
received.");
}

/*********************************************************************
* Method to write out when 5 is pressed
*********************************************************************/
void onDanFiftyResponse(dword methodCallHandle, dword messageResponseHandle
)
{
writeEx(0,1,"\nSUB2-SOME/IP: dan_limit 50 Code() method response
received.");
}
```

Event Handler for when the reset button is pressed

```
on sysvar_update SUB2::ResetFaultCode
{
byte dataBuffer[1];
if (@this == 1)     // prevent double-triggering with panel button controls
      {
dataBuffer[0]=0;
// set value of field content
SomeIpSetData(resetFaultCodeSetter,1,dataBuffer);
CheckForSomeIpError();

// call setter method
SomeIpCallMethod(resetFaultCodeSetter);
CheckForSomeIpError();
      }
}
```

Method to get the current danger limit value

```
/*********************************************************************
* Method to get the current Danger Limit Value
*********************************************************************/
void getDangerLimitMethod(DWORD danHandle, DWORD messageHandle)
{
    DWORD res;      // value of return parameter
    char data[1];

    // get the returned parameter values
    res = SomeIpGetData(messageHandle,elCount(data), data);
    CheckForSomeIpError();

    writeEx(0,1,"\nSUB2-SOME/IP: DangerPressure event data received: %d
",data[0]);

    @SUB2::DangerDsp = data[0];
}
```

Method when the r key is pressed to run the resetFaultCodeSetterMethod

```
/*********************************************************************
* When the r key is pressed the resetFaultCodeSetter Method is run
*********************************************************************/
on key 'r'
{
 byte dataBuffer[1];
      dataBuffer[0]=0;

  {
      // set value of field content
      SomeIpSetData(resetFaultCodeSetter,1,dataBuffer);
      CheckForSomeIpError();

      // call setter method
      SomeIpCallMethod(resetFaultCodeSetter);
      CheckForSomeIpError();
  }
}
```

Method for when the 4 key is pressed, to run the setDangerLimitLow method -40

```
/*********************************************************************
* When the key 4 is pressed the setDangerLimitLow method is run
*********************************************************************/
on key '4'
{
  byte dataBuffer[1];
       dataBuffer[0]=0;

  {
       // set value of field content
       SomeIpSetData(setDangerLimitLow,1,dataBuffer);
       CheckForSomeIpError();

       // call setter method
       SomeIpCallMethod(setDangerLimitLow);
       CheckForSomeIpError();
  }
}
```

Method for when the 5 key is pressed, to run the setDangerLimitHigh method -50

```
/*********************************************************************
* When the key 5 is pressed the setDangerLimitHigh method is run
*********************************************************************/
on key '5'
{
  byte dataBuffer[1];
       dataBuffer[0]=0;
  {
       // set value of field content
       SomeIpSetData(setDangerLimitHigh,1,dataBuffer);
       CheckForSomeIpError();

       // call setter method
       SomeIpCallMethod(setDangerLimitHigh);
       CheckForSomeIpError();
  }
}
```

Get the DangerLimit Value

```
/**********************************************************************
* when G is pressed, Current_dan_val_method is triggered.
**********************************************************************/
on key 'g'
{
    SomeIpCallMethod(Current_dan_val_method);
    CheckForSomeIpError();
}
```

```
/**********************************************************************
* Function to handle when the g button is pressed, it writes out the danger
value in the
* write window
**********************************************************************/
void OnDangerResponse(dword methodCallHandle, dword messageResponseHandle )
{
  DWORD res;         // value of return parameter
  char data[1];


  // get the returned parameter values
  res = SomeIpGetData(messageResponseHandle,elCount(data), data);
  CheckForSomeIpError();

  writeEx(0,1,"\nDASH-SOME/IP: Get Danger limit() method return value:  %d
",data[0]);

  @SUB2::DangerDsp = data[0];
}
```

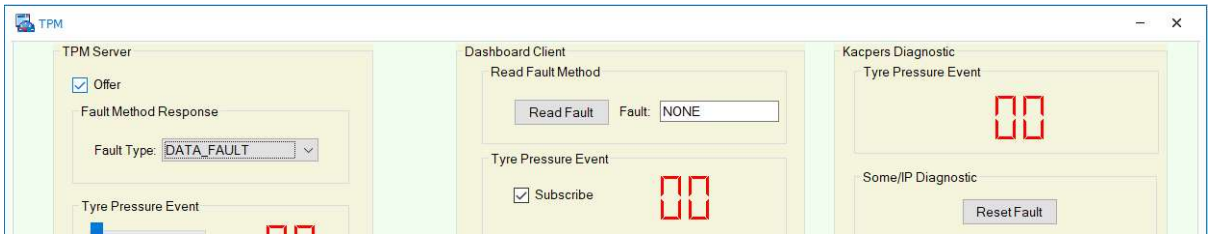
Results

Server Start

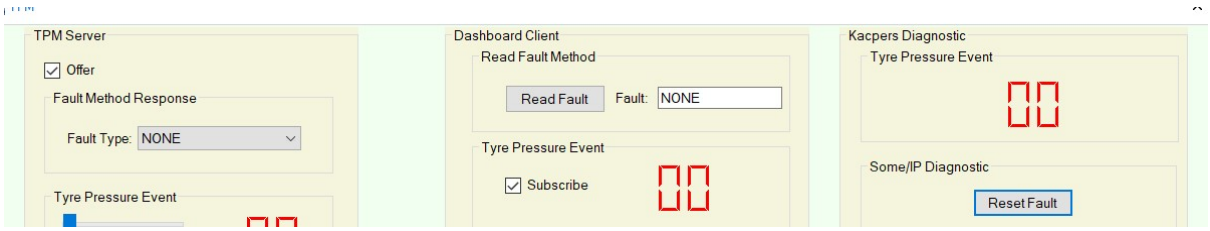
```
System      and animation factor = 1.
System    TPMS-SOME/IP: Server started.
```

Reset Button Pressed

Before



After

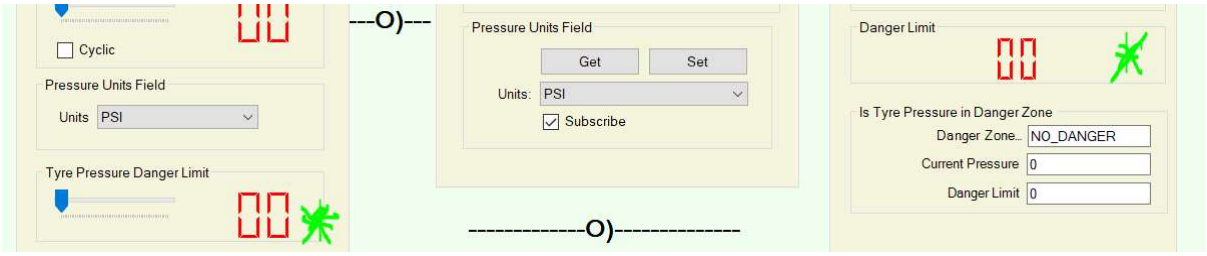


Write Window

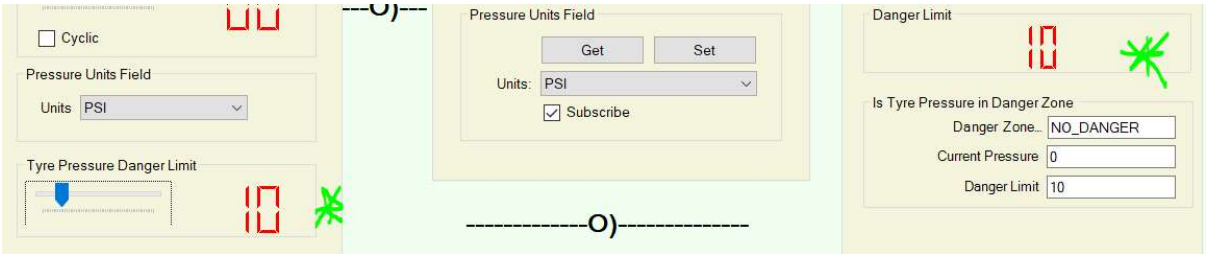
```
System    SUB2-SOME/IP: Reset Fault Code() method response received.
```

Tyre Pressure Danger Limit 0-60

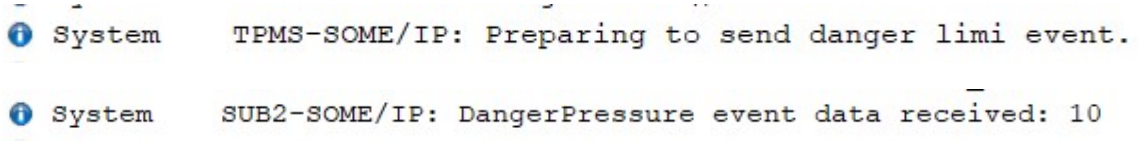
Before



After

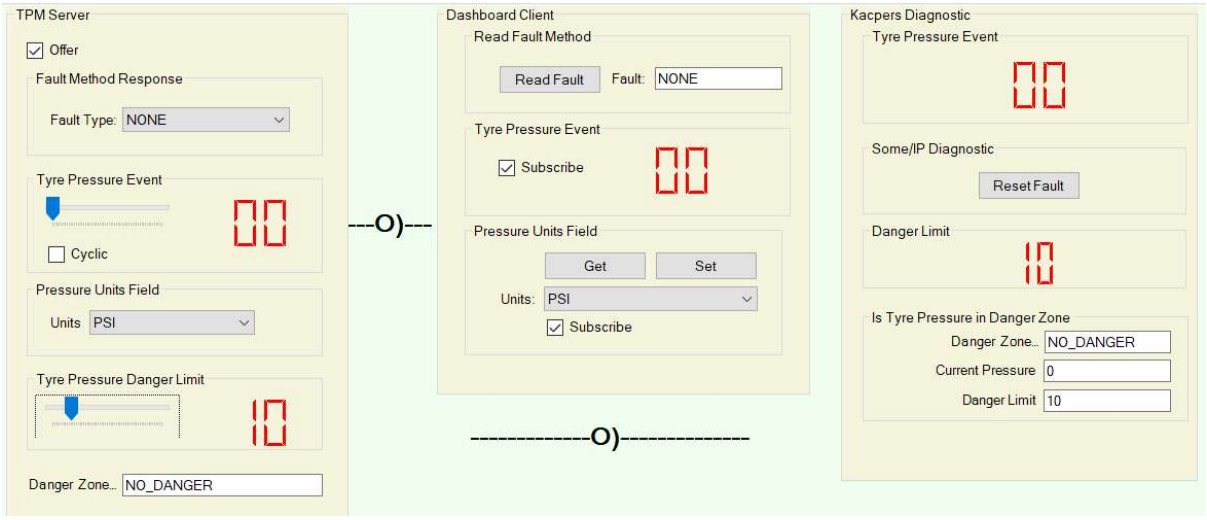


Write Window



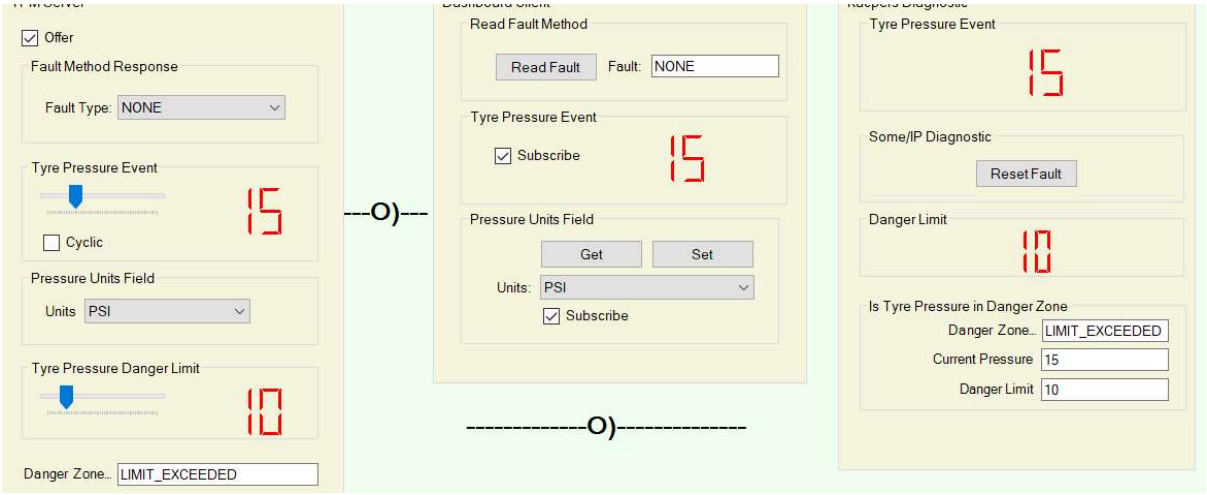
Danger limit Event - > When Pressure is bigger than Danger limit

Before



The Value of tyre pressure event is less than tyre pressure danger hence no danger is displayed, I will now change the value of tyre pressure to be greater than 10.

After



The value changed in the systems variable table from 0->1 i.e. from no danger to limit exceeded with the current pressure and danger limit visible, due to the fact that 15, tyre pressure is bigger than 10, the danger limit.

Write Window

- **Before (No Danger)**

```
System    DASH-SOME/IP: Danger zone() method return value:  0
```

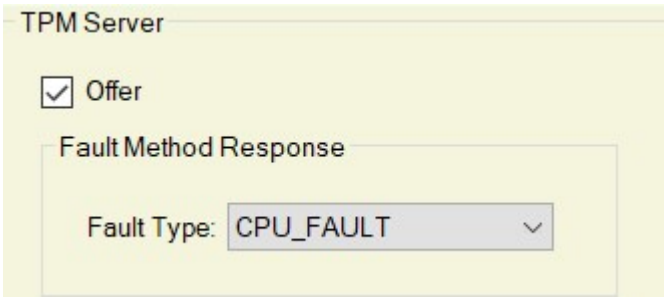
- **After (In Danger Zone)**

```
System    SUB-SOME/IP: tyrePressure event data received: 15
System    DASH-SOME/IP: Danger zone() method return value:  1
```

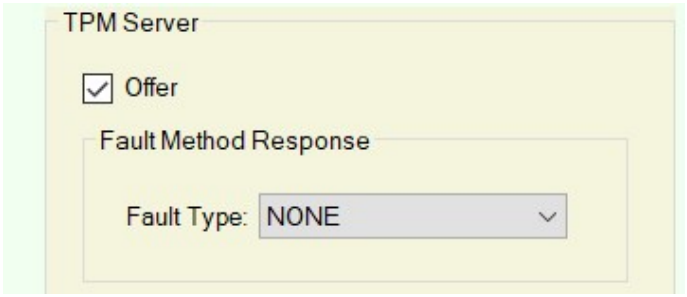
When R is pressed

When R is pressed, and the list is the last thing used, Canoe goes to Radio Fault due to it thinking and searching the list of possibilities that start with R, so that's Canoes bug, press anywhere on the GUI and then press R to make sure the list is not active for the to work properly.

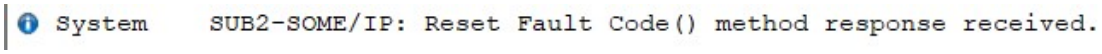
Before



After

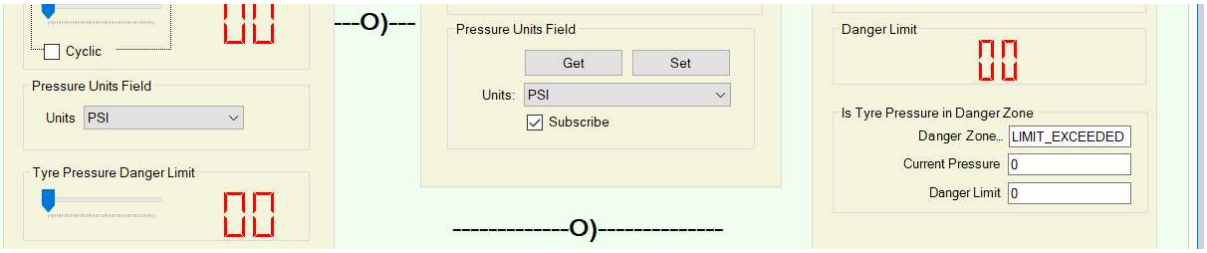


Write Window

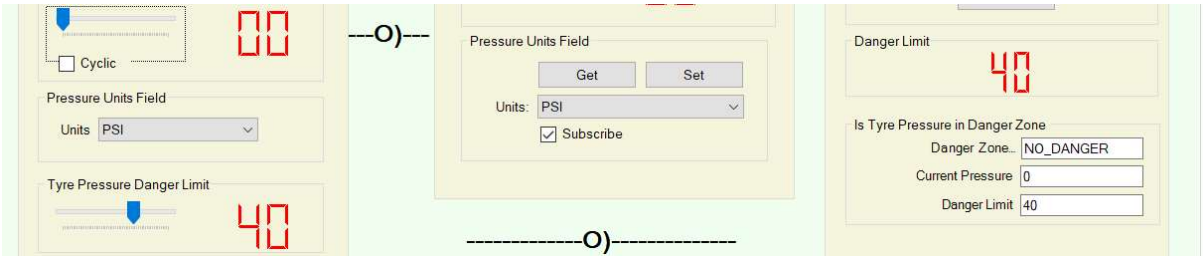


When 4 is pressed to set danger limit to 40

Before



After

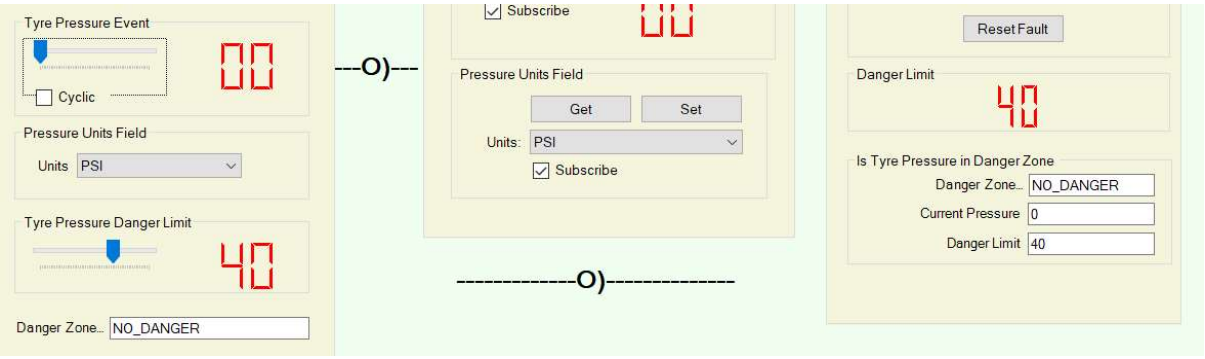


Write Window

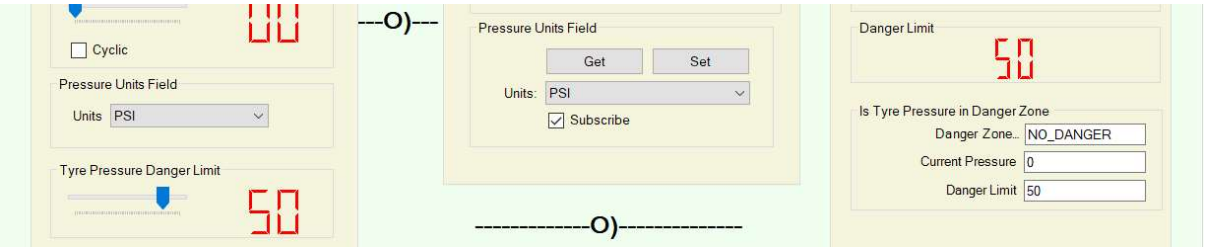
```
System    DASH-SOME/IP: Danger zone() method return value:  0
System    TPMS-SOME/IP: Preparing to send danger limi event.
System    TPMS-SOME/IP: Preparing to send danger zone value event.

System    SUB2-SOME/IP: dan_limit 40 Code() method response received.
System    SUB2-SOME/IP: DangerPressure event data received: 40
System    DASH-SOME/IP: Danger zone() method return value:  0
```

When 5 is pressed to set danger limit to 50

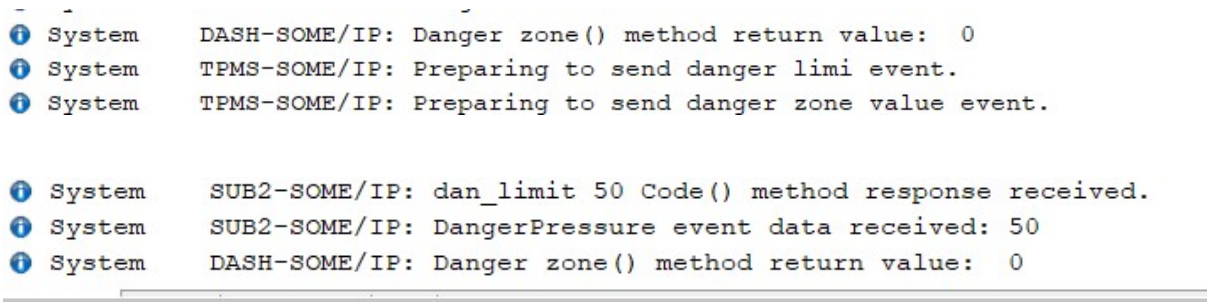
Before



After

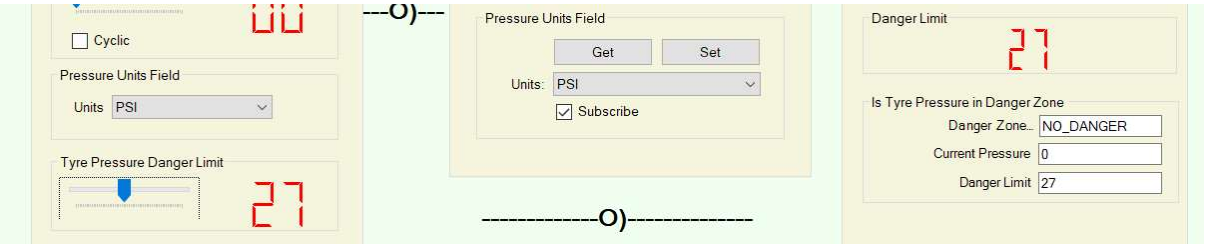


Write Window

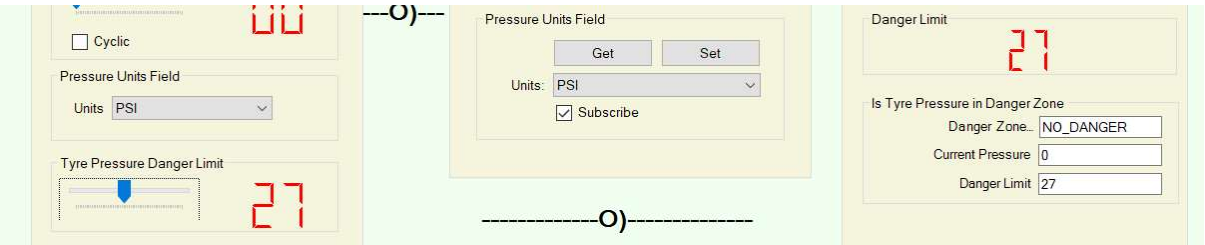


When G is pressed

Before



After



Write Window

```
System    TPMS-SOME/IP: Preparing to send danger limi event.
System    TPMS-SOME/IP: Preparing to send danger zone value event.
CAPL ...  46-0502 Informational in node DASH in SomeIP_IL.dll: received some
CAPL ...  46-0502 Informational in node DASH in SomeIP_IL.dll: received some
System    SUB2-SOME/IP: DangerPressure event data received: 27
System    DASH-SOME/IP: Danger zone() method return value:  0
System    DASH-SOME/IP: Get Danger limit() method return value:  27
```

When Value is changed

```
Source    Message
System    TPMS-SOME/IP: Preparing to send danger limi event.
System    TPMS-SOME/IP: Preparing to send danger zone value event.
CAPL ...  46-0502 Informational in node DASH in SomeIP_IL.dll: received some
CAPL ...  46-0502 Informational in node DASH in SomeIP_IL.dll: received some
System    SUB2-SOME/IP: DangerPressure event data received: 37
System    DASH-SOME/IP: Danger zone() method return value:  0
System    DASH-SOME/IP: Get Danger limit() method return value:  37
```

When Value is constant

```
Source    Message
System    DASH-SOME/IP: Get Danger limit() method return value:  37
System    DASH-SOME/IP: Get Danger limit() method return value:  37
System    DASH-SOME/IP: Get Danger limit() method return value:  37
System    DASH-SOME/IP: Get Danger limit() method return value:  37
System    DASH-SOME/IP: Get Danger limit() method return value:  37
System    DASH-SOME/IP: Get Danger limit() method return value:  37
System    DASH-SOME/IP: Get Danger limit() method return value:  37
```