STM32 DFL168A API

Table of Contents

1 Features Highlights

- Simply use our object DFL168A and its method to get motor data.
- Don't need to know OBD2 and J1939/J1708/J1587 protocol, don't need to read DFL168A data sheet, just need to know DFL168A pinout, so easily get vehicle parameters values in real time.
- Synchronous version can be used in simple situation, Or you use RTOS like FreeRTOS situation.
- Asynchronous version can be used in multi-task co-operation situation
- Can be used in all STM32 MCU.

2 Hardware

Please read DFL168A reference schematic in page 8 of DFL168A Datasheet or DFL168AM Pinout in Page 2 of DFL168AM Datasheet

For STM32, you can use one of UART1 to UART4, which have no flow control, have one stop bit, have no parity, have 8 bits Data. Baud Rate can be 57600 b/second.

Of cause, you can use other baud rate, however, you must set DFL168A Programmable parameter in address 1c by manual AT Command.

3 How to install API?

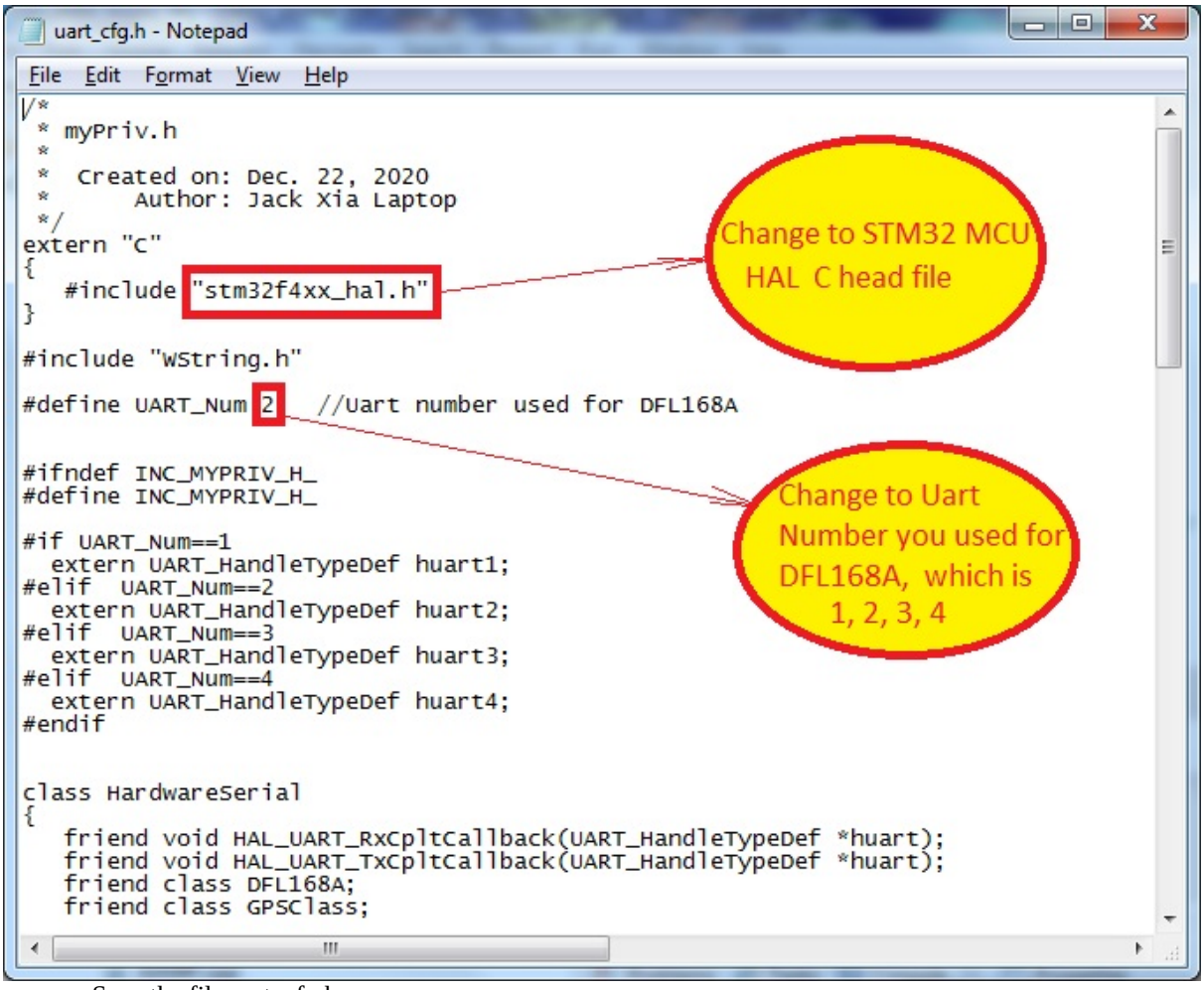
Install API steps:

Step1: Download API zip file : DFL168A_STM32_Sync.zip for synchronous version (http://dafulaielectronics.com/DFL168A_STM32_Sync.zip) or DFL168A_STM32_Async.zip for asynchronous version (http://dafulaielectronics.com/DFL168A_STM32_Async.zip)

Step2: unzip the source code to one folder

Step3: Enter folder "Inc" under source code, find file uart_cfg.h, change statement ' #include "stm32f4xx_hal.h" ' to the C HAL head file that your stm32 MCU used. Change number "2 " in the statement "#define UART_NUM 2" to the number your UART used.

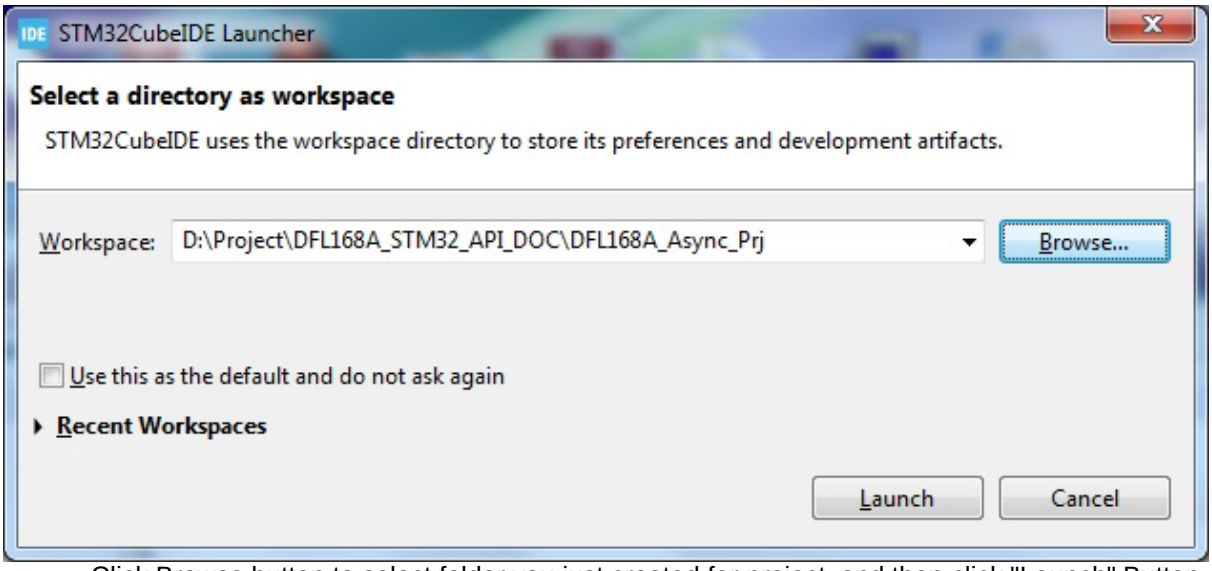
Please see screenshot below:



Save the file uart_cfg.h

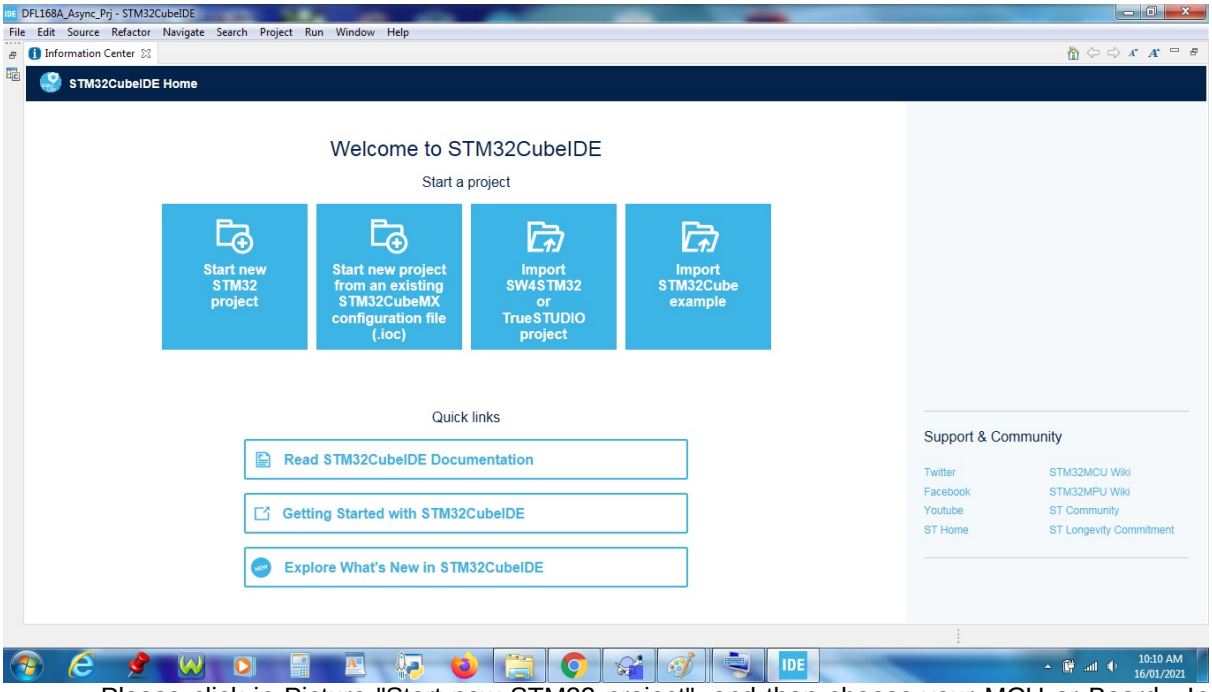
Step 4: This step 4 to step 9 are for setup STM32CubeIDE project. If you have existing STM32CubeIDE project, please directly go to step 10 .

Firstly we setup one project folder, for example , Folder D: \Project \DFL1568A_STM32_API_DOC\DFL168A_Async_Prj. You can use any folder name. Double click in icon STM32CubeIDE to run stm32cubeide.exe software, it will display window below:



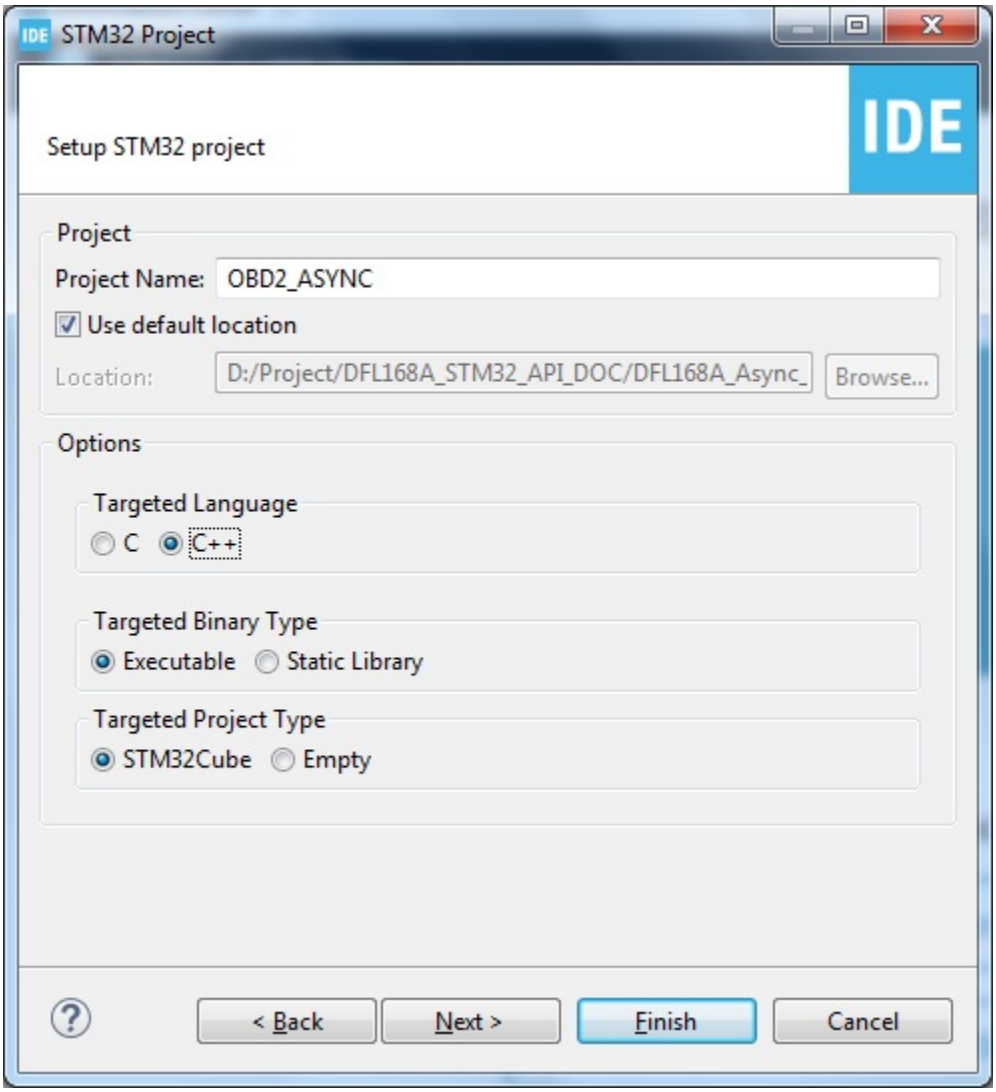
Click Browse button to select folder you just created for project, and then click "Launch" Button.

Step 5: You will see window below after step 4.

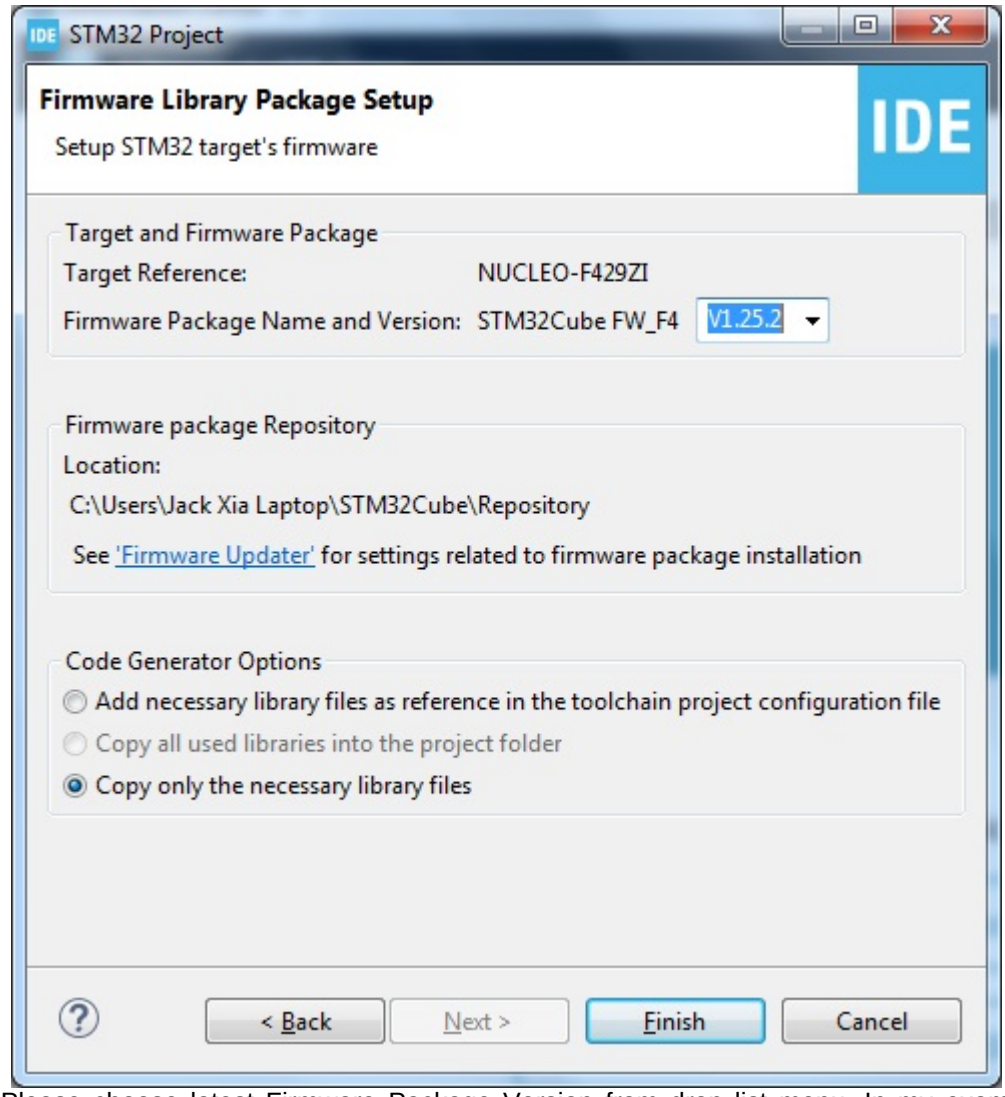


Please click in Picture "Start new STM32 project", and then choose your MCU or Board . In my example, it is Nucleo-F429ZI board. You may use your customized board with special MCU, You just choose your MCU in your case.And then click button "Next"

Step 6: You will see window below after step 5

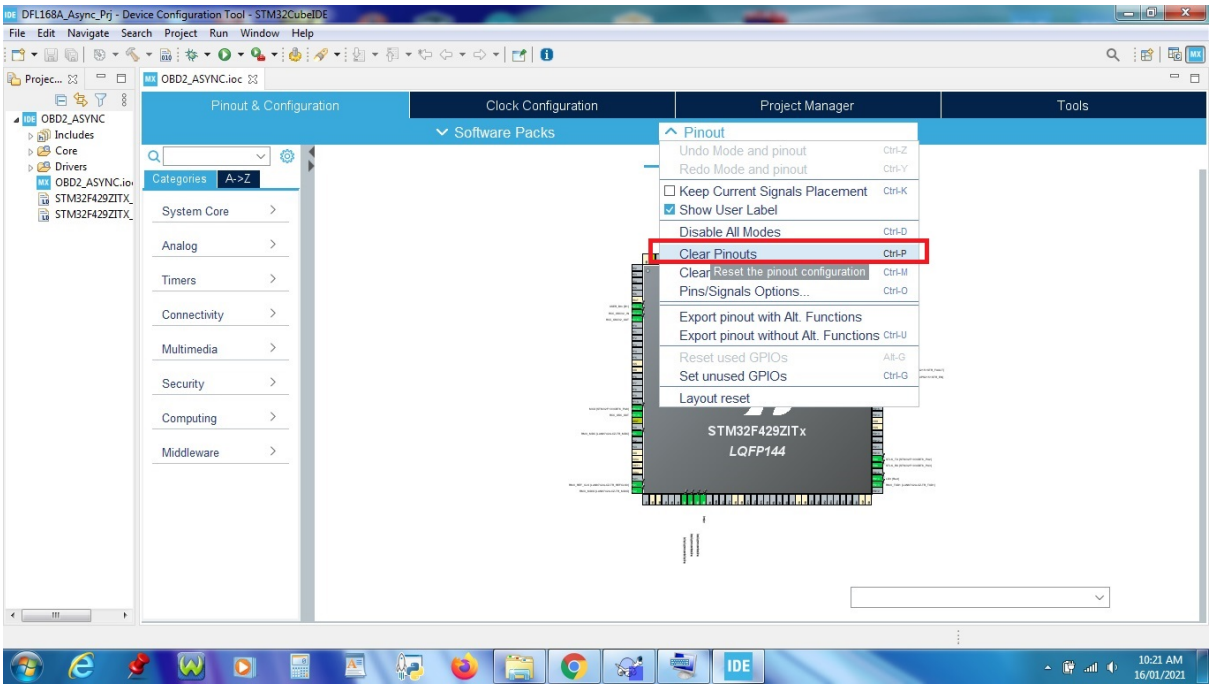


Please Type your Project Name , we used name "OBD2_ASYNC", you can used any name. Targeted Language must select C++. You maybe don't know C++, and only know C language. Don't worry it, you still use C to write application except our DFL168A API with C++. And then please click Button "Next", you will see window below:



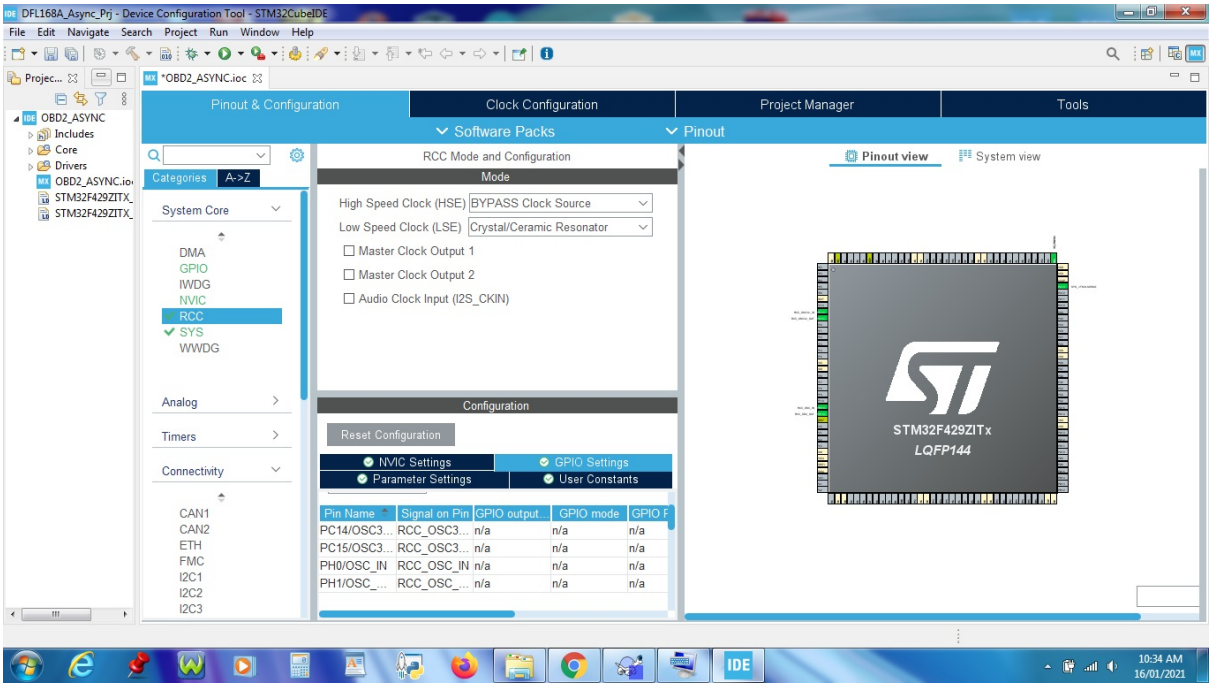
Please choose latest Firmware Package Version from drop list menu. In my example, it is v1.25.2. Click button "Finish". If any dialog occurs, just use default "Yes" Button to continue.

Step 7: Please click menu "Pinout/Clear Pinouts" which is shown below:

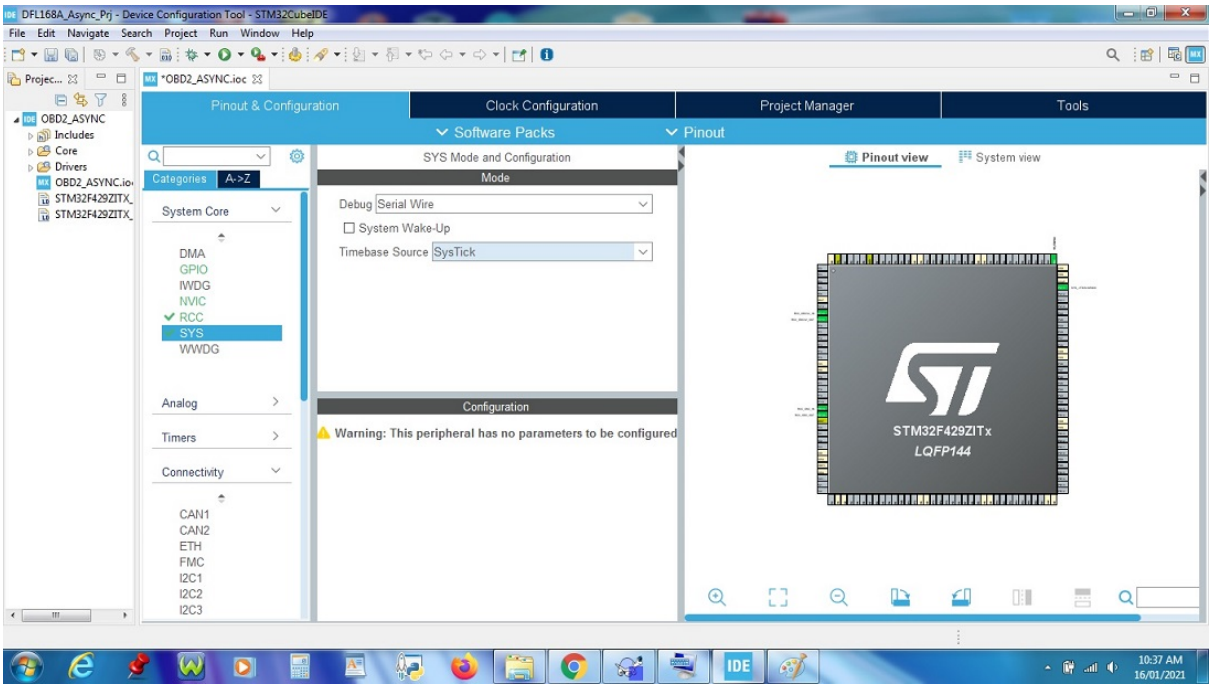


If any dialog occurs, just use default "Yes" Button to continue.

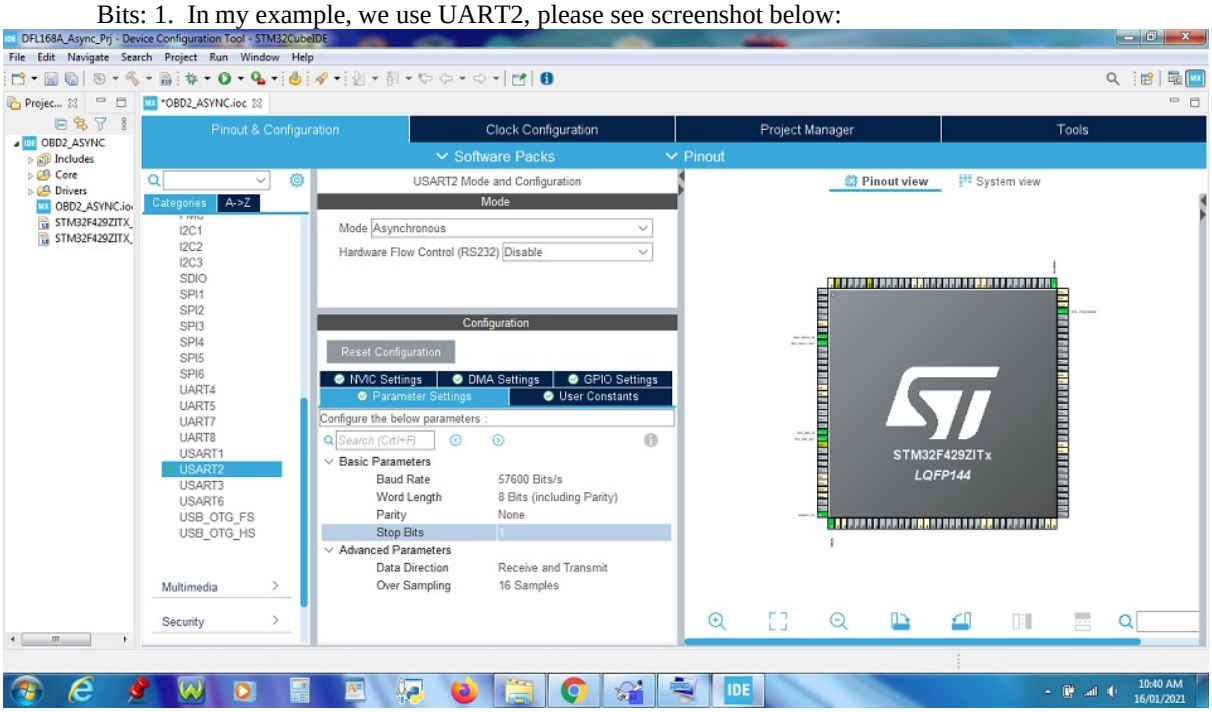
Step 8: Please config RCC according to your case. In my example, we use configuration below:



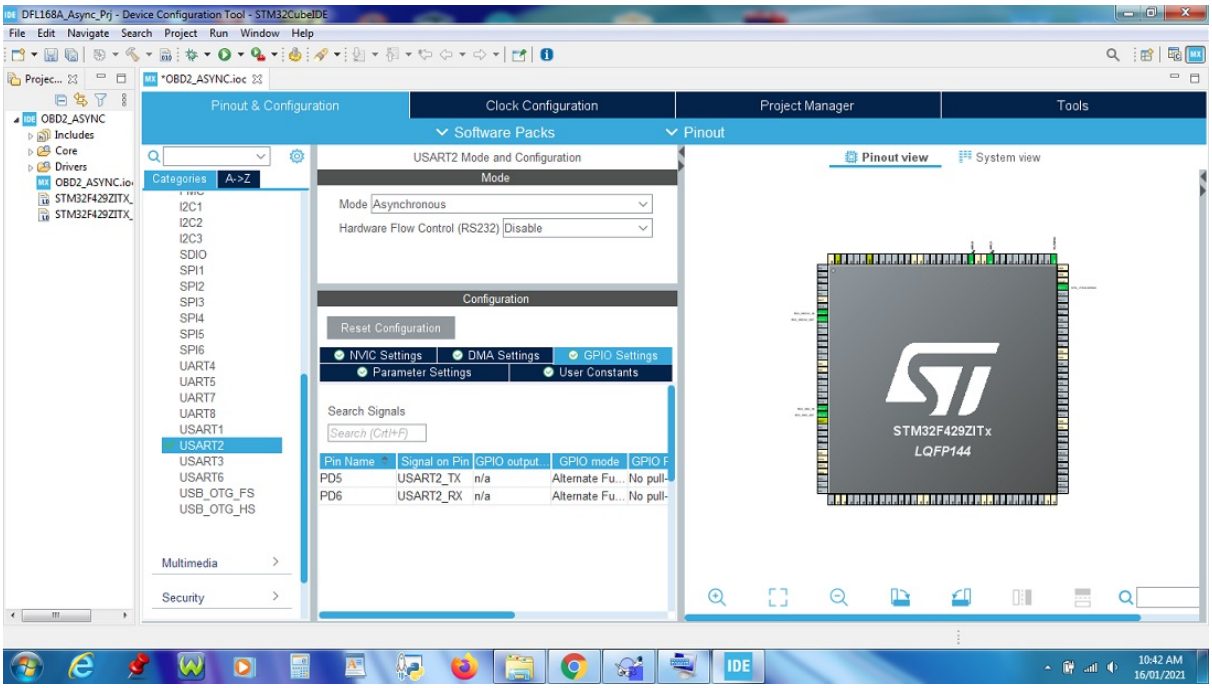
Step 9: Please choose TimeBase Source as SysTick , See screenshot below:



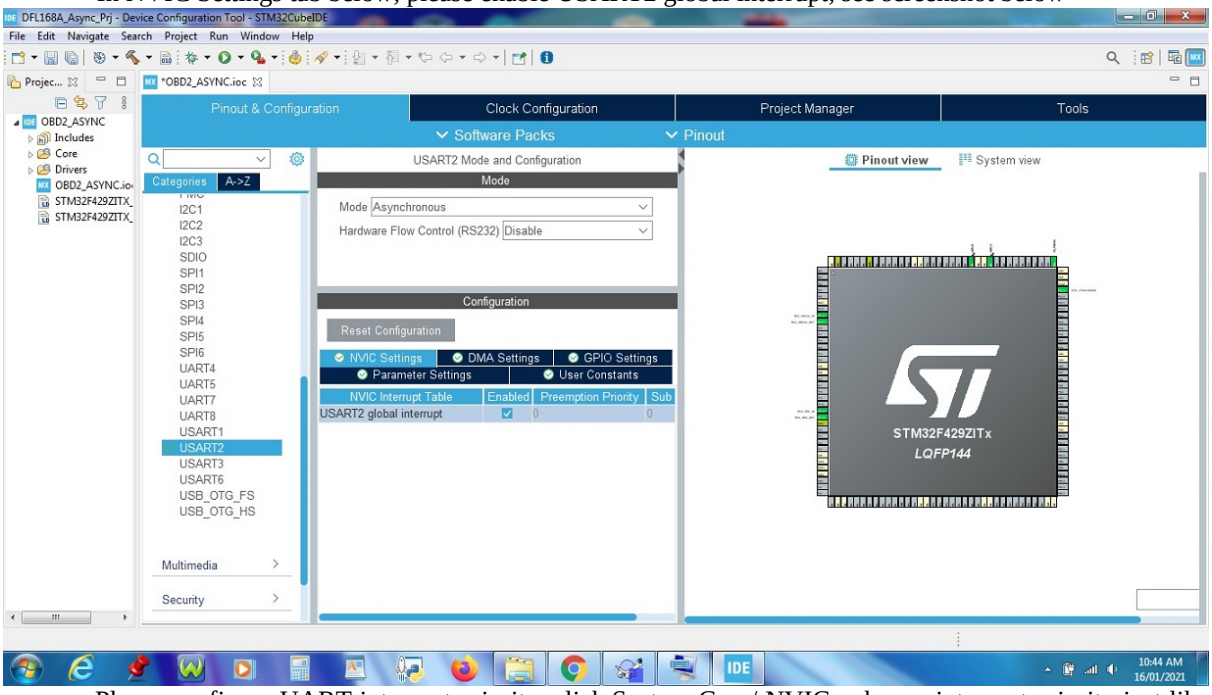
Step 10: Please configure UART you used for DFL168A. Mode: Asynchronous, Hardware Flow Control (RS232): Disable,. In Parameter Settings tab, Buad Rate: 57600, Word Length: 8, Parity: None, Stop Bits: 1. In my example, we use UART2, please see screenshot below:



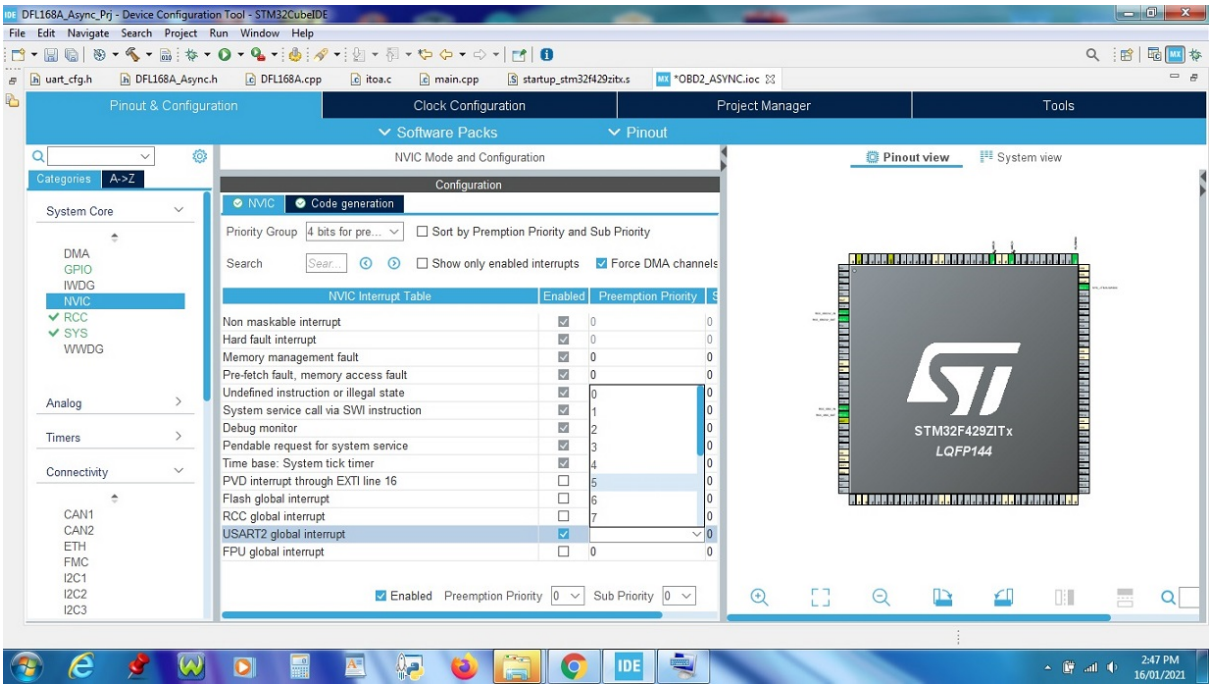
In GPIO tab below, please configure correct Pin for TX and RX, see screenshot below:



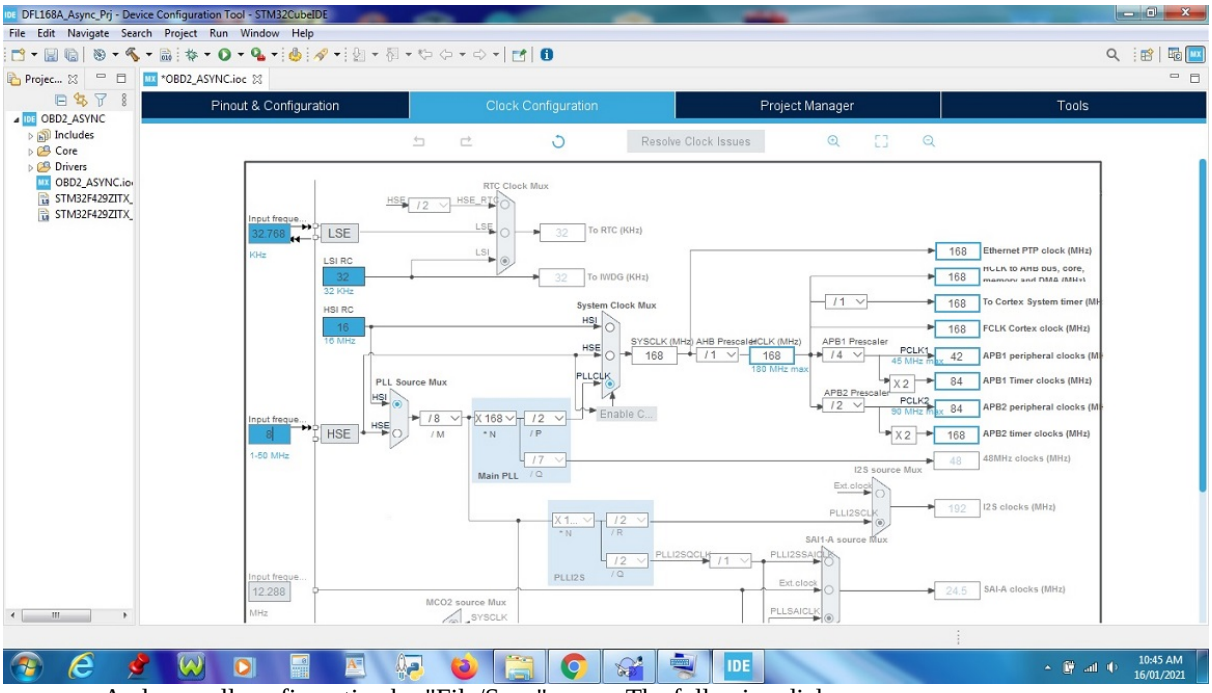
In NVIC Settings tab below, please enable USART2 global interrupt, see screenshot below



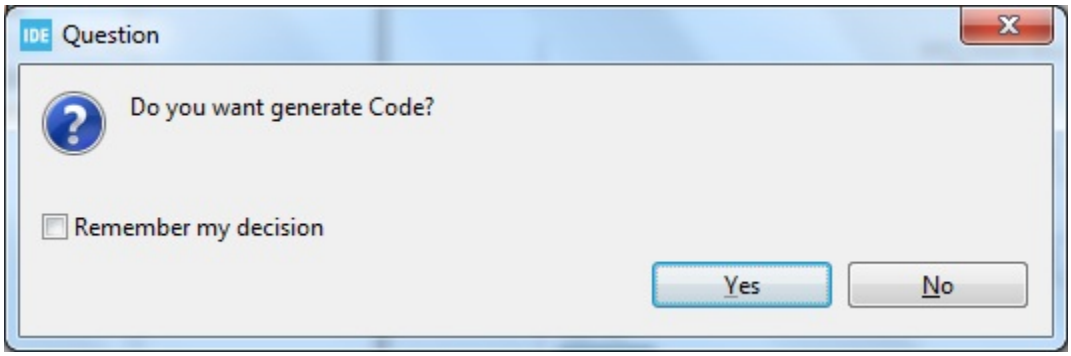
Please configure UART interrupt priority, click System Core/ NVIC , change interrupt priority just like screenshot below:



Step 11: You can change clock as you need , see screenshot below:

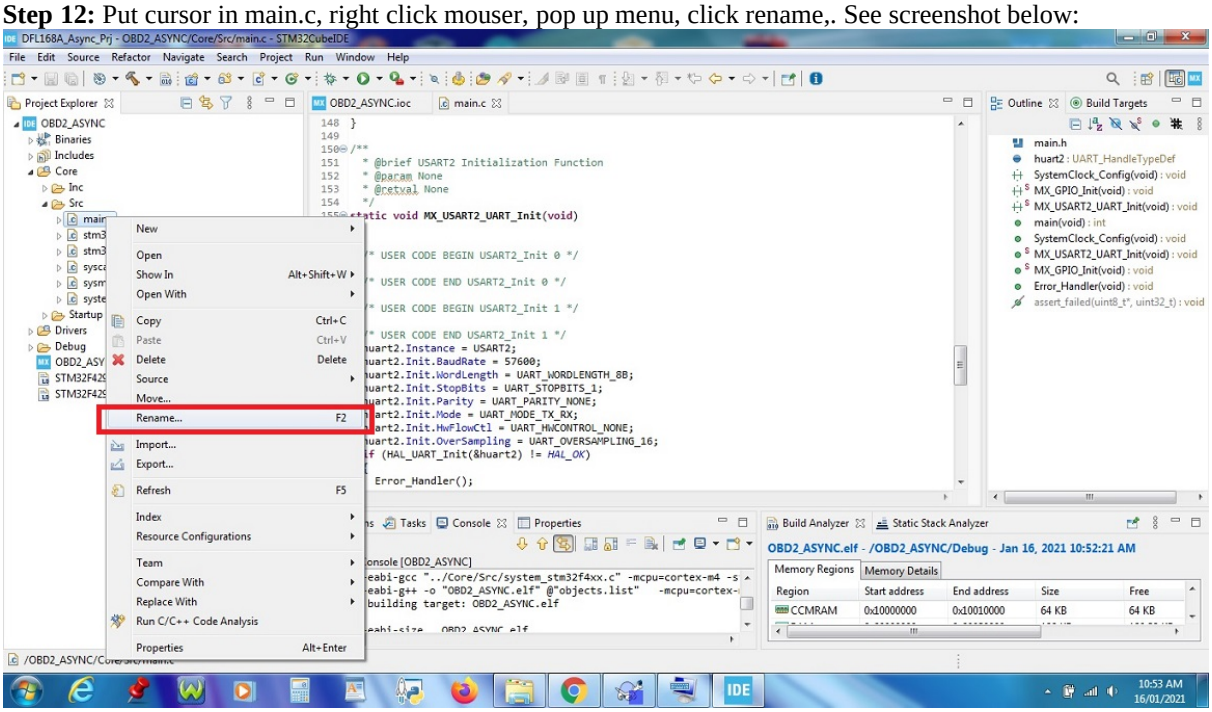


And save all configuration by "File/Save" menu. The following dialog occurs,

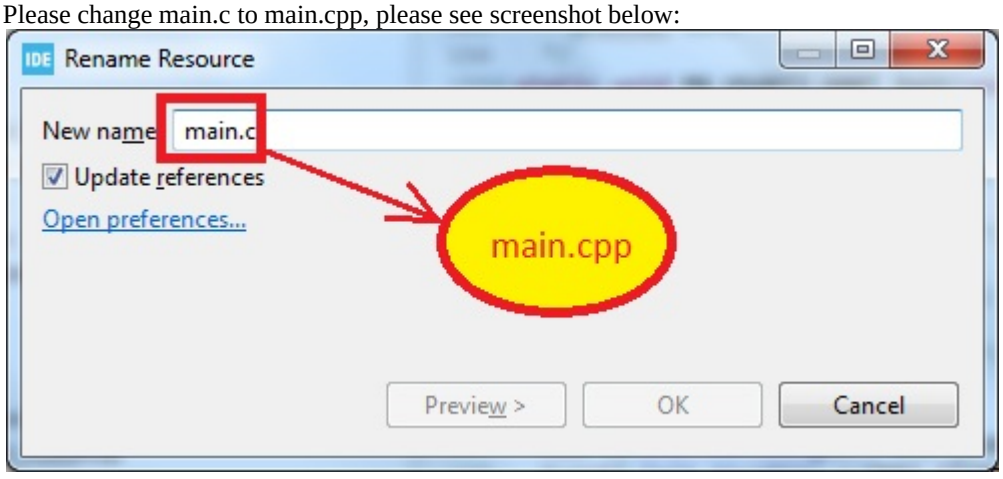


Just click "Yes" to generate Code.

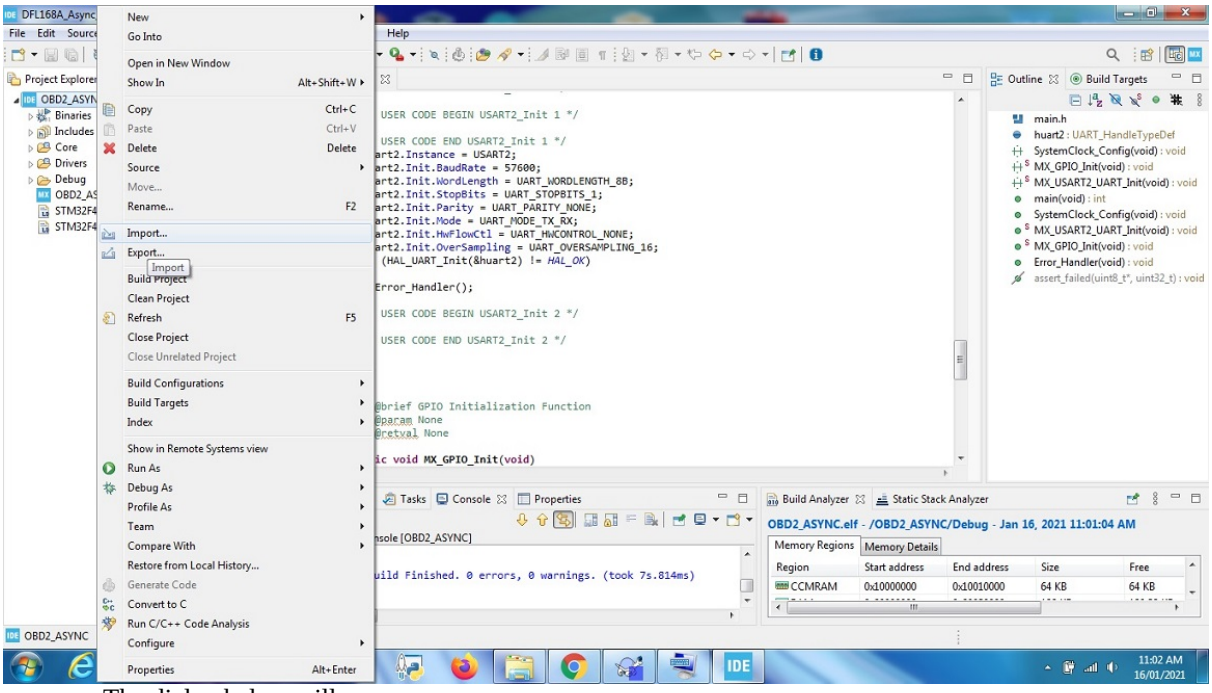
Step 12: Put cursor in main.c, right click mouser, pop up menu, click rename,. See screenshot below:



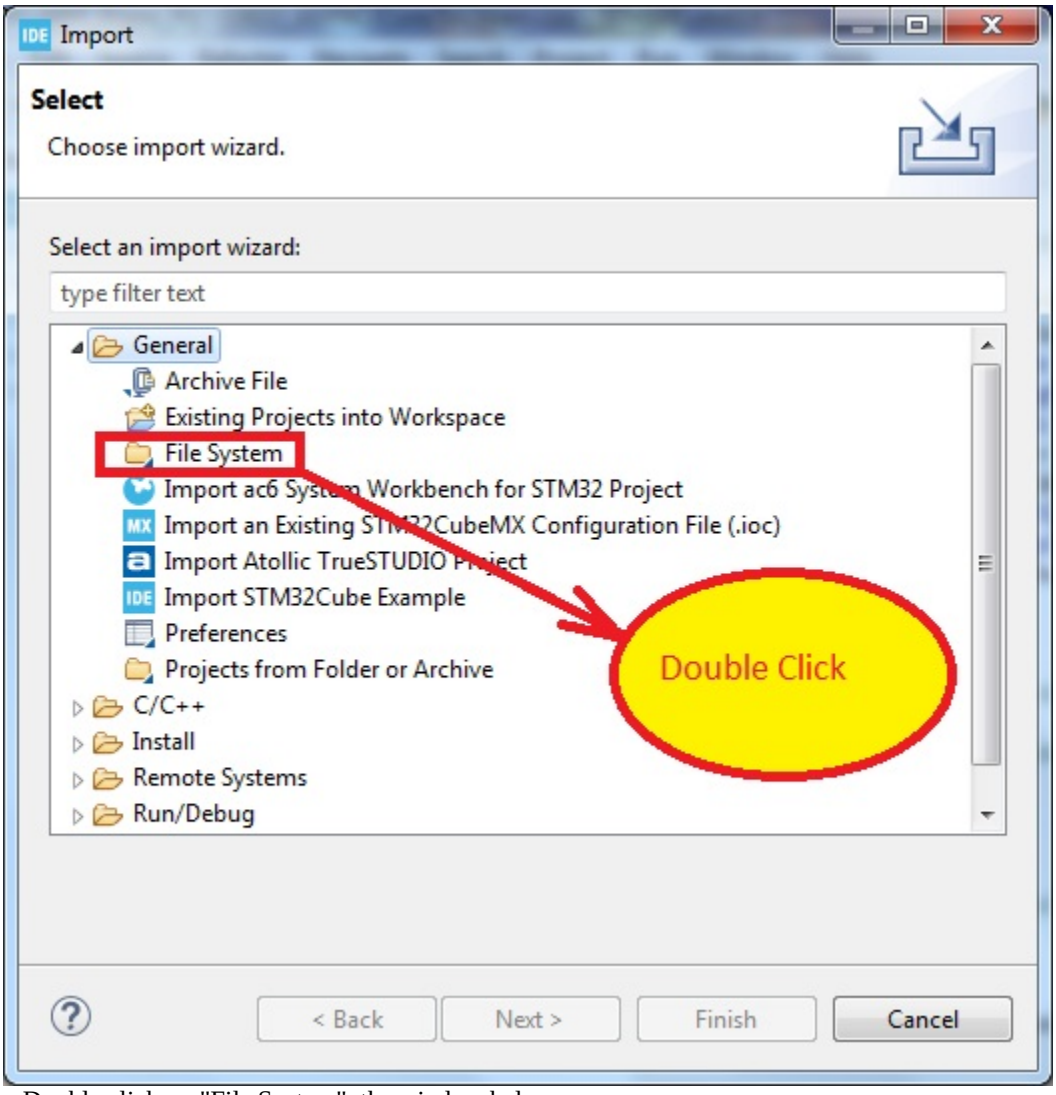
Please change main.c to main.cpp, please see screenshot below:



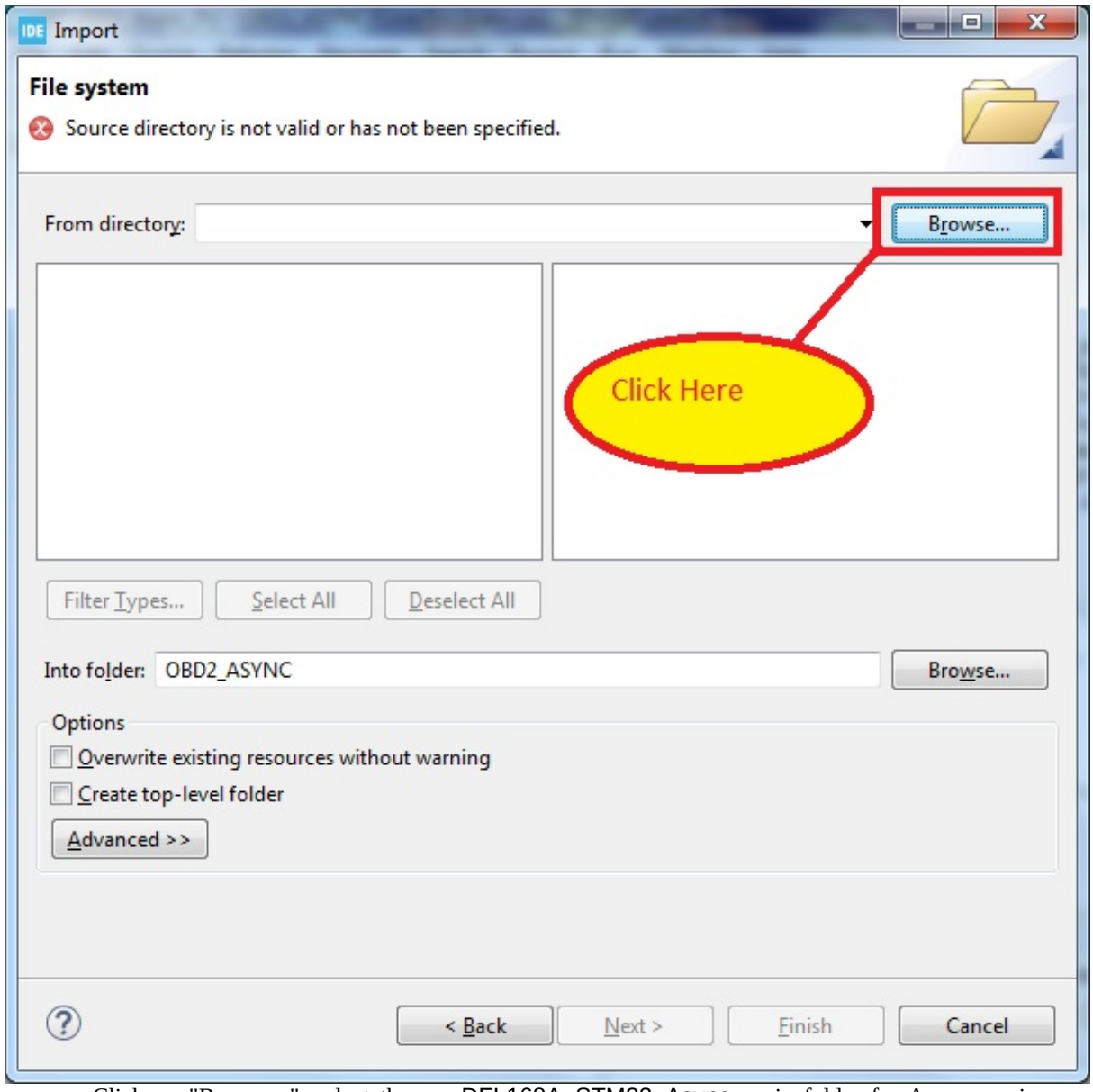
Step 13: Put cursor in project name, right click mouser, pop up menu, click Import,. See screenshot below:



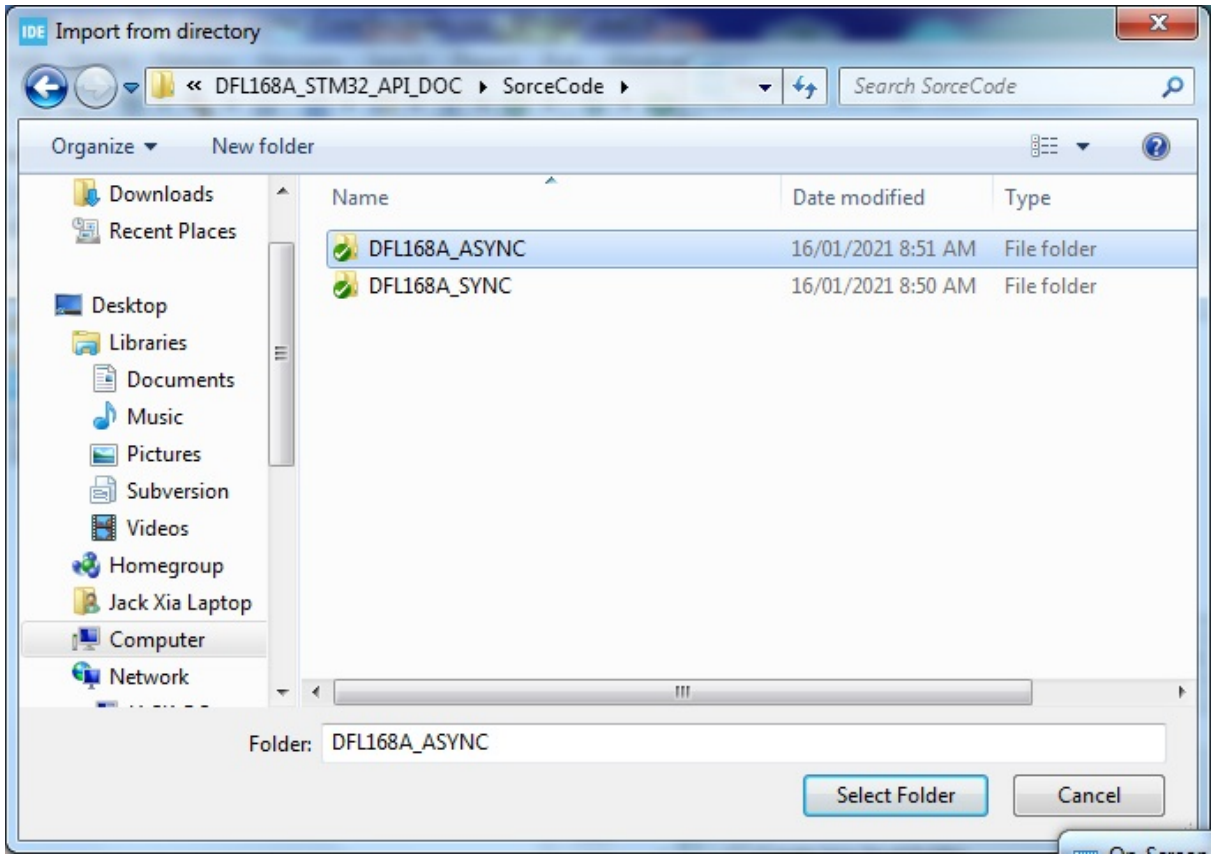
The dialog below will occur.



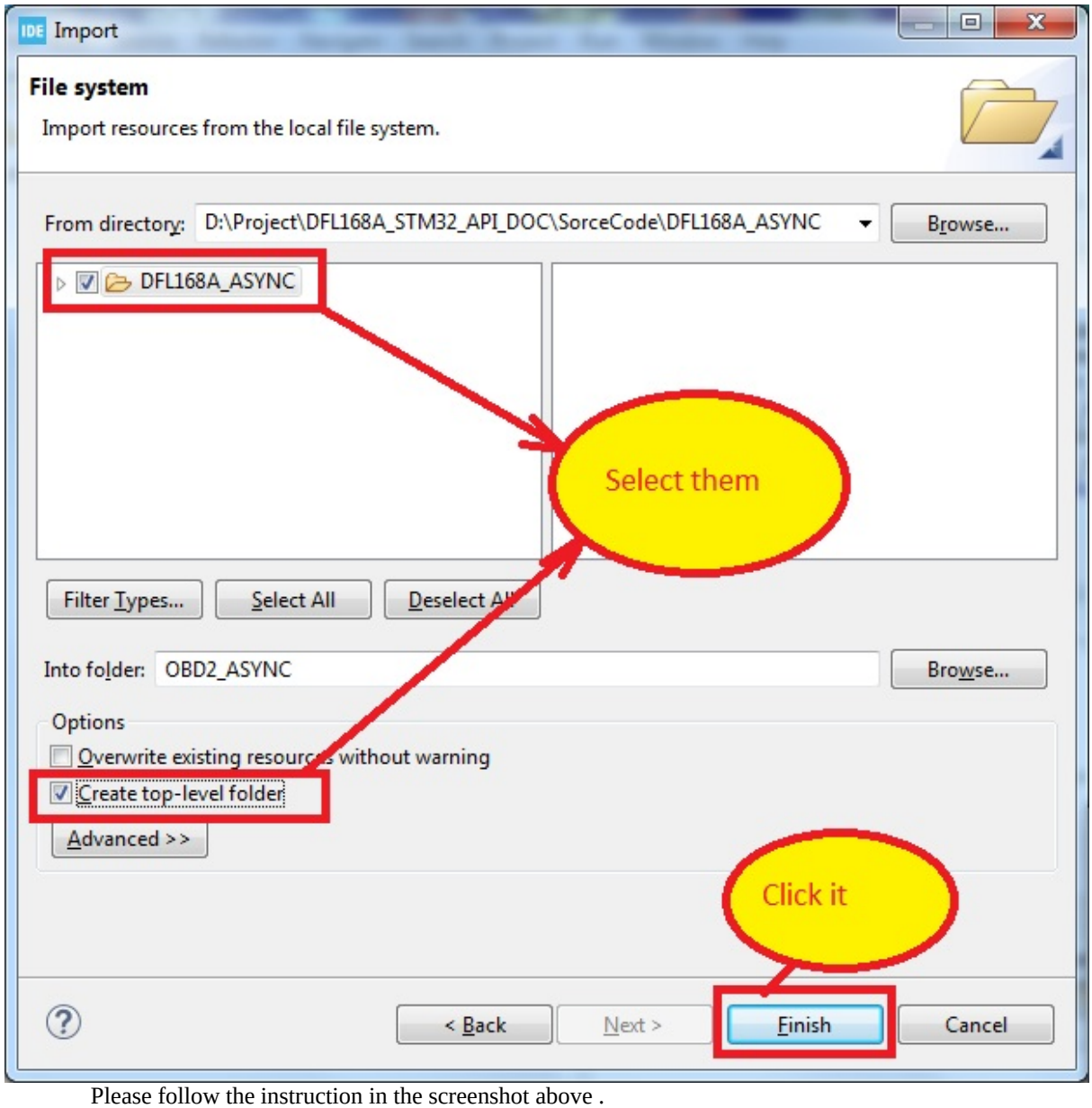
Double click on "File System", the window below occurs.



Click on "Browse...", select the our DFL168A_STM32_Async unzip folder for Async version or DFL168A_STM32_Sync unzip folder for Sync version , and select DFL168A_ASYNC folder for Async version or DFL168A_SYNC folder for Sync version. Our example is "DFL168A_ASYNC" folder which is Async version. Please see screenshot below:

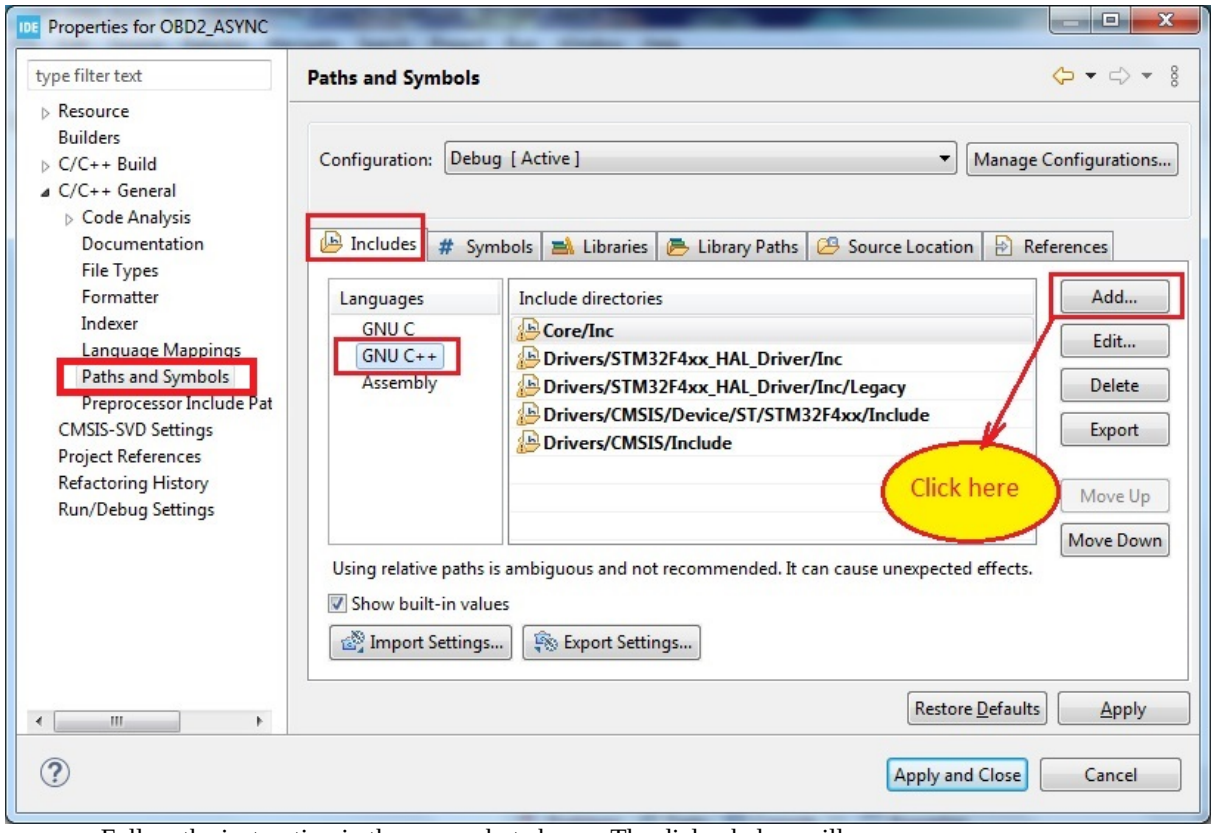


Click on "Select Folder" , the window below occurs.

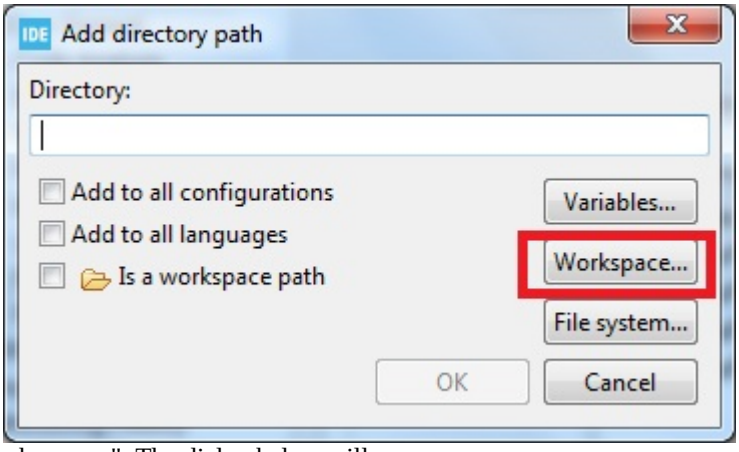


Please follow the instruction in the screenshot above .

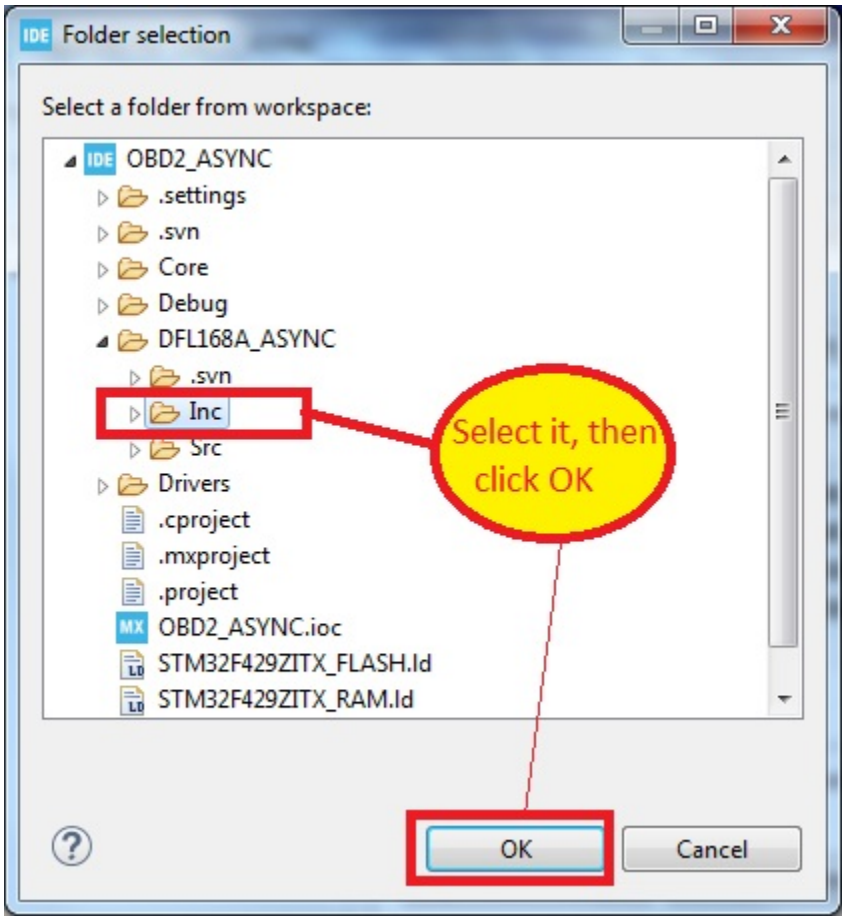
Step 14: We already add our DFL168A API source code. However Compiler does not know it. Now the following steps will make compiler knows them. Please click on menu "Project/Properties"
We will see dialog below:



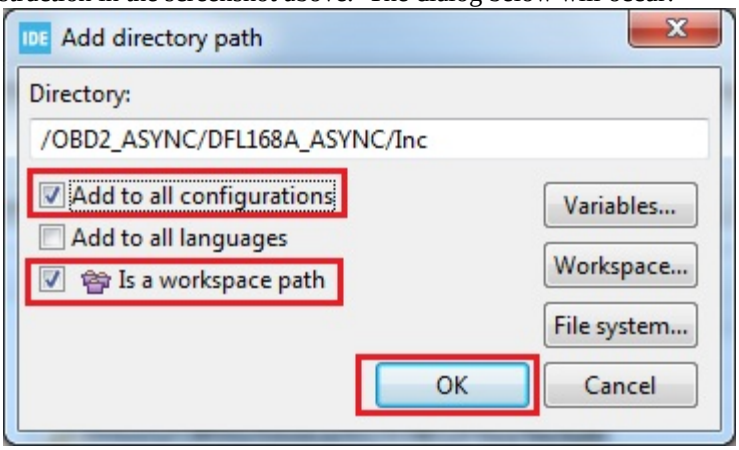
Follow the instruction in the screenshot above. The dialog below will occur:



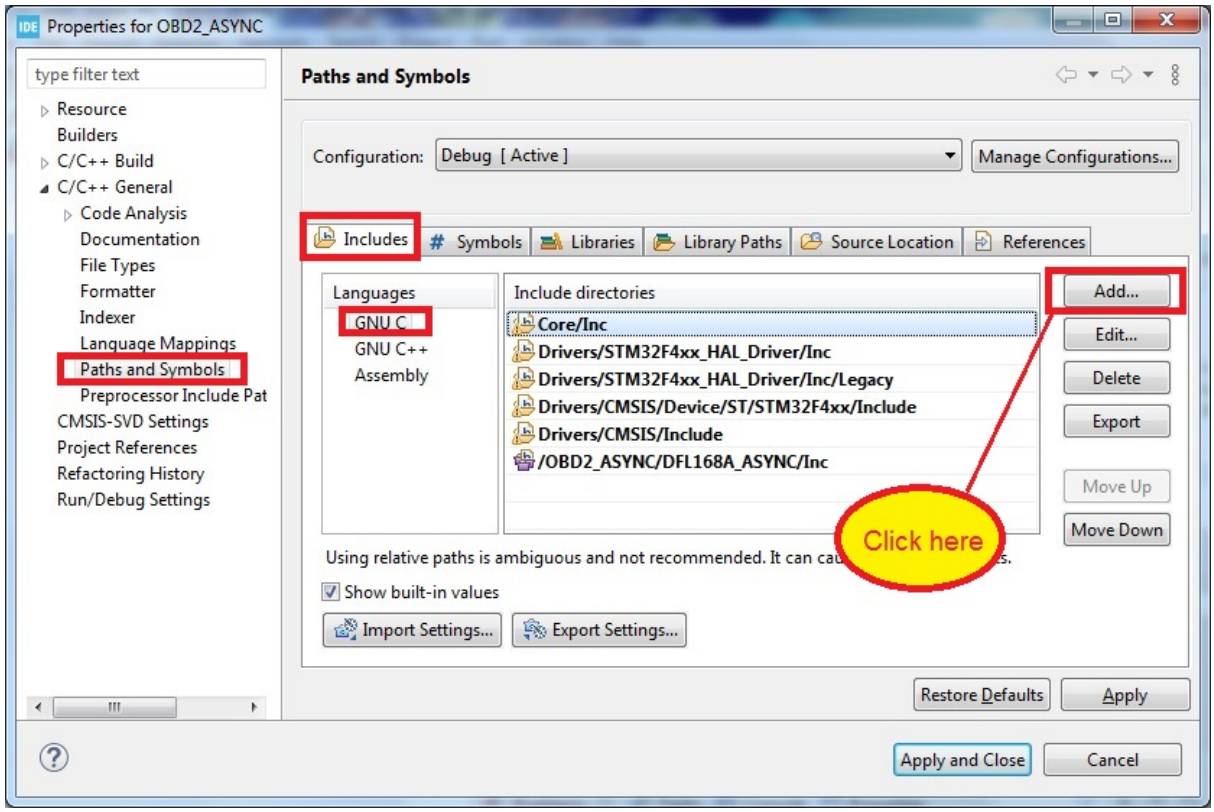
Click on "Workspace...". The dialog below will occur:



Follow the instruction in the screenshot above. The dialog below will occur:

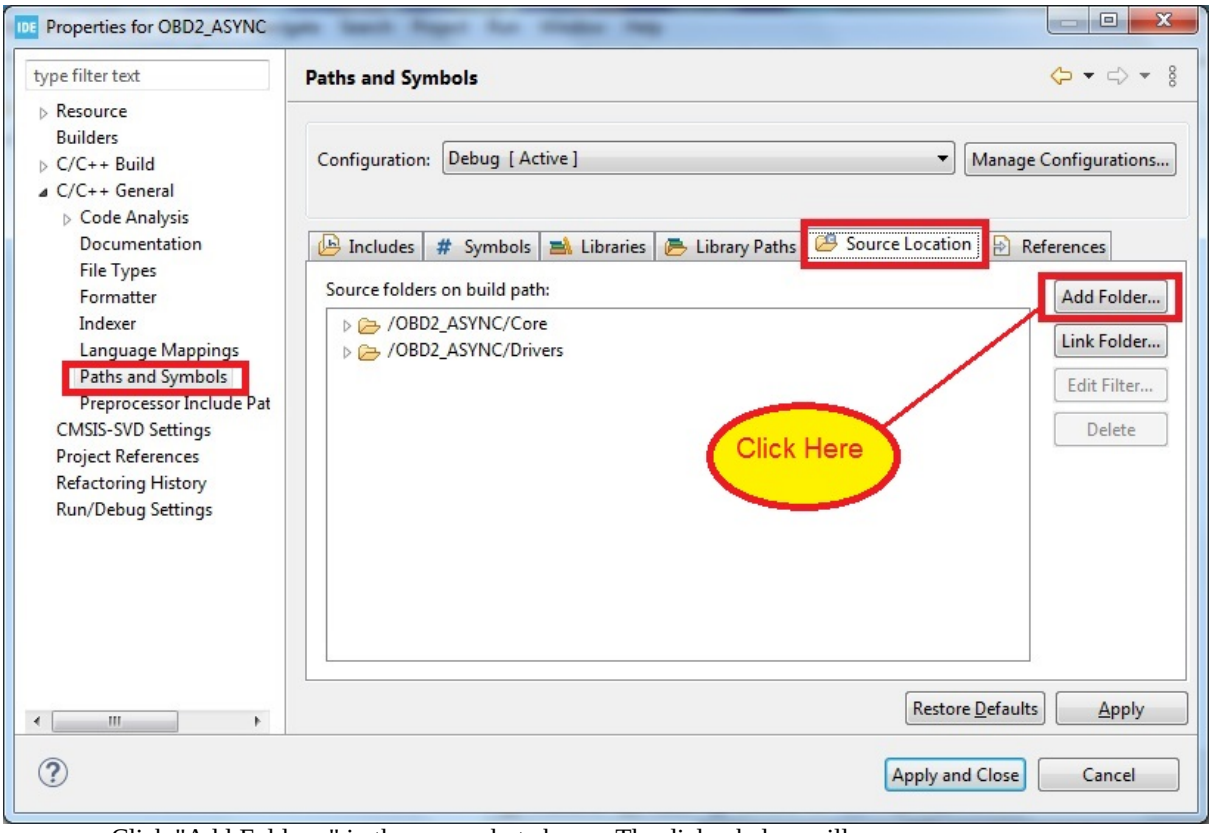


Step 15: In previous Step, we let Compiler know C++ head file location for our DFL168A API. In this step, we will let Compiler know C head file location for our DFL168A API. Follow the instructions in the screenshot below.

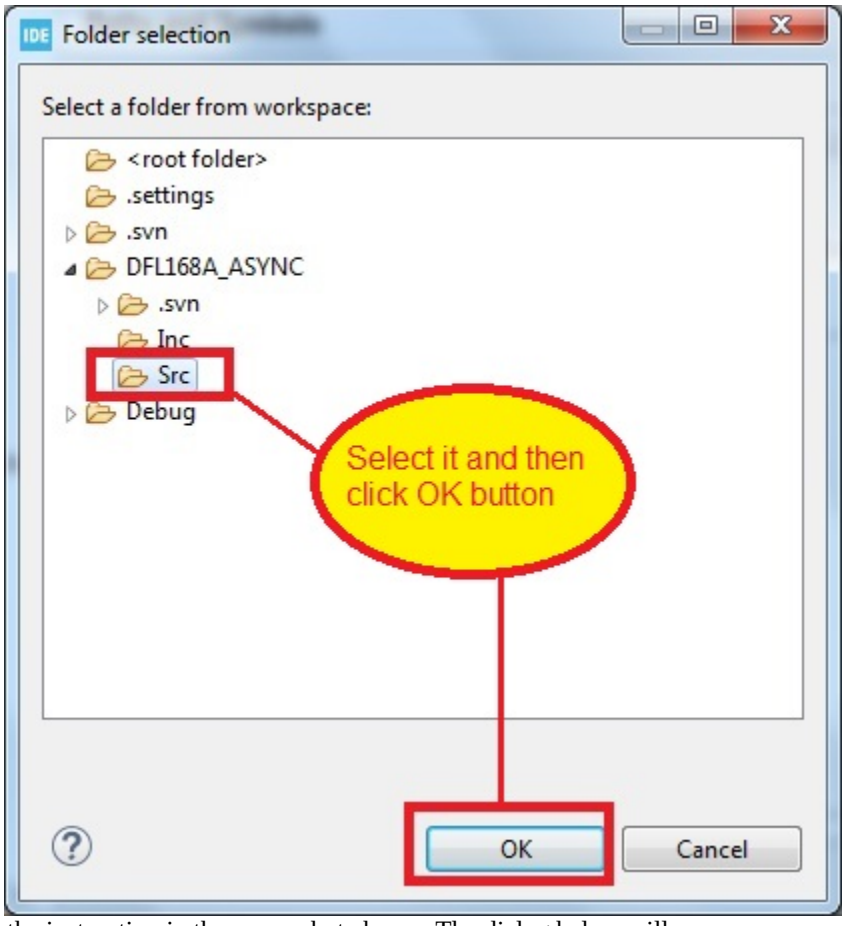


The following operations are the same as step 14.

Step 16: In step 14 and step 15, we let Compiler know C++ and C head file location for our DFL168A API. In this step, we will let Compiler know source file location for our DFL168A API. Please see dialog below:

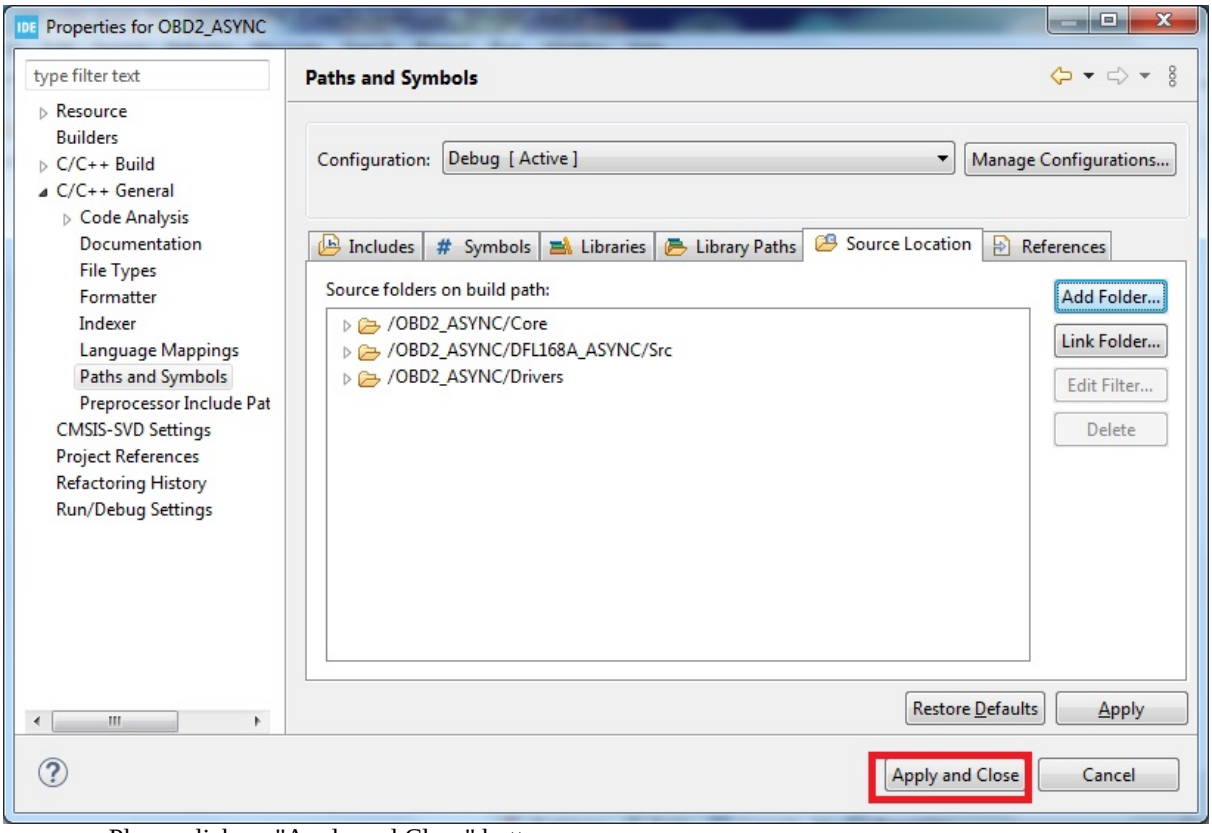


Click "Add Folder..." in the screenshot above. The dialog below will occur:



Follow the instruction in the screenshot above. The dialog below will occur:

"



Please click on "Apply and Close" button

4 How to use API?

Notes: 1. Our DFL168A used One UART port, and It used function

*HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) , and*

*HAL_UART_TxCpltCallback(UART_HandleTypeDef *huart).*

If you used other UARTs to do other job (which is not for DFL168A) for your application, and if you need function HAL_UART_RxCpltCallback and/or HAL_UART_TxCpltCallback, please don't write function HAL_UART_RxCpltCallback and HAL_UART_TxCpltCallback. Please Modify function void HAL_UART_RxCpltCallback and void HAL_UART_TxCpltCallback in file uart_cfg.cpp

*For example, Function void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart), the original code is below:*

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
  HardwareSerial * mySeri;
   #if UART_Num==1
   mySeri=&Serial1;
```

```
 if (&huart1==huart)
 #elif UART_Num==2
  mySeri=&Serial2;
  if (&huart2==huart)
 #elif UART_Num==3
  mySeri=&Serial3;
  if (&huart3==huart)
 #elif UART_Num==4
  mySeri=&Serial4;
  if (&huart4==huart)
 #endif
 {
            mySeri->RxCplt=1;

 }
}
```

*If your UART7 will do other job which is different from DFL168A, your Function void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) will become:*

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
  HardwareSerial * mySeri;
  #if UART_Num==1
   mySeri=&Serial1;
   if (&huart1==huart)
  #elif UART_Num==2
   mySeri=&Serial2;
   if (&huart2==huart)
  #elif UART_Num==3
   mySeri=&Serial3;
   if (&huart3==huart)
  #elif UART_Num==4
   mySeri=&Serial4;
   if (&huart4==huart)
  #endif
  {
            mySeri->RxCplt=1;
```

```
    }
      //Add your code  ---begin
      if (&huart7==huart)
      {
      }
      //Add your code  ---End
}
```

2 *We have String class which is different from Standard C++ string class. Our String class is the same as Arduino String . The string class source code is from Arduino. So please see reference : https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/*

Firstly, you should have statement " #include "DFL168A_Async.h" between "/* USER CODE BEGIN Includes */" and "/* USER CODE END Includes */" in main.cpp if you use DFL168A Async version API.

Otherwise, you should have statement " #include "DFL168A_Sync.h" between "/* USER CODE BEGIN Includes */" and "/* USER CODE END Includes */" in main.cpp if you use DFL168A Sync version API.

Secondly, you should define global object in main.cpp by the following statement:

```
DFL168A myDFL168A(/* Protocol Name*/ J1939_PROTOCOL,
                  /* Protocol Timeout (ms)*/1000,
                  /* Protocol Baudrate (bit/second)*/250000,
                  /* Baudrate for GPS or DEV1*/ 9600,
                  /* Timeout (ms) for GPS or DEV1*/500);
```

The above statement must be between "/* USER CODE BEGIN PV */" and "/* USER CODE END PV */"

For this construction above, you can read Construction Function

Of cause, if you use ISO15765, you can simply use statement "DFL168A myDFL168A;" which will use default ISO15765 Protocol.

In function" int main(void) " in main.cpp file , you must have statement below between "/* USER CODE BEGIN 1 */" and "/* USER CODE END 1 */":

```
/* USER CODE BEGIN 1 */
bool Vehicle_OK;
byte result;  //if Async version
//Other variable definition statements such as "String Vin;"  (These variables hold the vehicle Data)
/* USER CODE END 1 */
```

In function" int main(void) " in main.cpp file , you must have statement below between "/* USER CODE BEGIN 2 */" and "/* USER CODE END 2 */":

```
Vehicle_OK=myDFL168A.begin(true, false);
```

This statement above is the last statement before the code enter " while(1) endless loop". The above statement will be running for around 5 seconds , and get value of Vehicle_OK. The true value means that DFL168A IC is OK and DFL168A succeed in communication with STM32 MCU. The false value means that DFL168A IC is not OK and DFL168A fail in communication with STM32.

Inside main " while(1) endless loop", the first statement is added below:

```
myDFL168A.SerialPoll();   //must add it, it is very important
```

The above statement is used polling Uart data.

Inside main " while(1) endless loop" , after above statement , you can put any statement for get motor data, For example, you want to get VIN number, you can write code below for synchronous version:

```
if (myDFL168A.J1939.getVIN(Vin))
{
    //Action code for getting Vin successfully
}
else
{
    //Action code for failing to get Vin
}
```

If you use asynchronous version, you can write code below:

```
result=myDFL168A.J1939.getVIN(Vin);
if (WAITING==result)  continue;
if (SUCCESS==result)
{
//Action code for getting Vin successfully
}
else
{
//Action code for getting Vin successfully
}
```

If you use asynchronous version and you want to get more different vehicle data, you should use State Machine. For example , if you want VIN , Vehicle Speed, Ambient temperature, and Barometric Pressure, you can add statements between "/* USER CODE BEGIN 1 */" and "/* USER CODE END 1 */":

```
/* USER CODE BEGIN 1 */
        String Vin;
```

```
    float VehicleSpeed;
    float BarometricPressure;
      int temperature;
    byte StateMachine=0;
    byte result;
/* USER CODE END 1 */
```

Inside main " while(1) endless loop", the first statement is added below:

```
myDFL168A.SerialPoll();   //must add it, it is very important
```

Inside main " while(1) endless loop" , after above statement , you can put the following statements below:

```
if (Vehicle_OK)
{
    switch(StateMachine)
    {
    case 0:
        result=myDFL168A.J1939.getVIN(Vin);
        if (WAITING==result) break;
        if (SUCCESS==result)
        {
            StateMachine=1;
        }
        else
        {
            StateMachine=1;
        }
        break;
    case 1:
        result=myDFL168A.J1939.PGN65269.refresh();
        if (WAITING==result) break;
        if (SUCCESS==result)
        {
            if (myDFL168A.J1939.PGN65269.getBarometricPressure(BarometricPressure))
            {
                // The code for using BarometricPressure
            }
            else
            {
                //BarometricPressure value is not available
            }
```

```
            if (myDFL168A.J1939.PGN65269.getAmbientTemp(temperature))
            {
                //The code for using "Ambient Air Temperature "  :  temperature
            }
            else
            {
                //temperature value is not available
            }
        }
        else
        {
            //PGN65269 is not available
        }
        StateMachine=2;
        break;
    case 2:
        result=myDFL168A.J1939.PGN65265.refresh();
        if (WAITING==result) break;
        if (SUCCESS==result)
        {
            if (myDFL168A.J1939.PGN65265.getWheelBasedVehicleSpeed(VehicleSpeed))
            {
                //The code for using "Vehicle Speed ":  VehicleSpeed
            }
            else
            {
                //VehicleSpeed value is not available
            }
        }
        else
        {
            //PGN65265 is not available
        }
        StateMachine=3;
        break;
    case 3:
        result=myDFL168A.J1939.clearDTC();
```

```
            if (WAITING==result) break;
            if (SUCCESS==result)
            {
                //Clear DTC success
            }
            else
            {
                //fail to Clear DTC
            }
             StateMachine=0;
        }
    }
```

You can read all example in " Your source code location\DFL168A_Sync_examples" for synchronous version. And in "Your source code location\DFL168A_Async_examples" for asynchronous version. All examples use NUCLEO-F429ZI board and use USART2 communicate with DFL168A.

If you use other board or other stm32 MCU, you must change example code.

5 DFL168A synchronous version library

DFL168A Library has a class "DFL168A", and DFL168A class has inner class: J1939 class, J1708 class, ISO15765 class and GPS class. For J1939 class, it consists of lots of items. These items are PGN65267 class, PGN65262 class,..., PGN57344 class.

If you want some vehicle data, you should know which PGN it belongs to. And firstly, you should call its refresh method of PGN. After that, you can use the PGN class's method to get vehicle data. All methods are blocked, so all method will running and exit only when the methods finish.

For J1708 class, there are lots of methods which will give you vehicle data. These methods are blocked.

For ISO15765 class, there are lots of methods which will give you vehicle data. These methods are blocked.

For GPS class, there are methods which will give you GPS data. These methods are blocked.

5.1 Constant

For protocol name, we have the following constant in the DFL168A library:

AUTO_PROTOCOL---------- This is ISO15765 protocol, and DFL168A will select correct specific ISO15765 protocol automatically, such as 11bits or 29 bits and baud rate.

ISO15765_11_500_PROTOCOL----------This is ISO15765 protocol with 11 bits CAN ID and 500Kbps baud rate.

ISO15765_29_500_PROTOCOL----------This is ISO15765 protocol with 29 bits CAN ID and 500Kbps baud rate.

ISO15765_11_250_PROTOCOL----------This is ISO15765 protocol with 11 bits CAN ID and 250Kbps baud rate.

ISO15765_29_250_PROTOCOL----------This is ISO15765 protocol with 29 bits CAN ID and 250Kbps baud rate.

J1939_PROTOCOL----------This is J1939 protocol, and baud rate will be decided by your construction function.

J1708_PROTOCOL----------This is J1708/J1587 protocol,

5.2 Members

Class DFL168A has the following members:

J1939

J1708

ISO15765

GPS

Above members are actually object. We will explain these inner object later.

5.3 Methods

```
DFL168A( );
bool begin(bool intrude,bool Fast);
void end();
bool getOnewireID(byte (&ID)[7]);
bool getDIN(int portNo);
void setDOUT(int portNo, bool Value);
float getAnalog(); //0.0 to 999.00
void beginTransparentSerial();
void endTransparentSerial();
bool setExitTransparentKey(byte EndTransparentChar);
bool serialPortAvailable();
bool setSleepDelay(unsigned int SleepDelaySeconds);
```

5.3.1 Construction Function

DFL168A(byte currProtocol, int Timeout, long J1939Baudrate, long DEV_Baudrate, int DEV_timeout);

Description

You must use this construction function to declare a global DFL168A object. It will specify which Serial Port of Arduino used for DFL168A, and which protocol you use, and baud rate of DEV1 of DFL168A, and related time out value.

Syntax

DFL168A *YourObjectName*(currProtocol, Timeout, J1939Baudrate, DEV_Baudrate, DEV_timeout);

Parameters

currProtocol : the second parameter, byte type, this is input parameter. It tells DFL168A which protocol you select. It can be constant "AUTO_PROTOCOL", "ISO15765_11_500_PROTOCOL", "ISO15765_29_500_PROTOCOL" , "ISO15765_11_250_PROTOCOL", "ISO15765_29_250_PROTOCOL", "J1939_PROTOCOL", and "J1708_PROTOCOL"

Timeout: the third parameter, int type, this is input parameter.It actually sets up time out for DFL168A vehicle protocol. Its unit is ms. For example, we know one vehicle data will be broadcast every 2 seconds, so we can set up time out= 2000ms. It will cause that all block methods will be waiting for maximum 2000+50ms in order to get vehicle data.

J1939Baudrate: The fourth parameter, long type, this is input parameter.It set up J1939 baud rate. Unit is "bps". If you didn't use J1939, this parameter can be any value.

DEV_Baudrate: the fourth parameter, long type, this is input parameter. It is baudrate of DFL168A DEV1 serial port. Usually DEV1 serial port of DFL168A connects GPS module, so in this case, it will be GPS module baud rate. Of cause, DEV1 can be connected any other serial port device in the transparent mode of DFL168A. You can use beginTransparentSerial() method to enter transparent mode. And then you can use HardWSerial to access serial port device.

DEV_timeout: the sixth parameter, int type, this is input parameter. It actually sets up time out (ms) for GPS module which is connected to DEV1 serial port of DFL168A. If GPS module send out NMEA-183 sentence every 1 seconds, you should let DEV_timeout=1000

Returns

Nothing

Notes: If you use default construction (No any parameters) , It means ISO15765 Protocol, Timeout:500ms, DEV baud rate : 9600, DEV1 Timeout: 500ms

5.3.2 begin Method

bool begin(bool intrude,bool Fast);

Description

DFL168A.begin(intrude,Fast) will config DFL168A IC and check whether DFL168A IC is OK. If config succeed, it will return true, otherwise return false. This method will run as long as 5 seconds for setting DFL168A IC.

Syntax

DFL168A.begin(intrude,Fast);

Parameters

intrude: the first parameter, bool type, this is input parameter. It denotes DFL168A will send request to vehicle when its value is true. And DFL168A won't send request to vehicle for J1708/ J1939 protocol when its value is false, so DFL168A only uses broadcast of vehicle data.

Fast: the second parameter, bool type,this is input parameter. It won't send any config data to DFL168A if its value is true. So in this situation, you should setup DFL168A by yourself via hyper-terminal. Library will send all config data to DFL168A if its value is false. You don't need take care of anything. Library takes care of everything for you.

Returns

bool

5.3.3 end Method

void end();

Description

DFL168A.end() will release Arduino serial port which is used by DFL168A, and You can use

DFL168A.begin(intrude,Fast) again only after you run this method.

Syntax

DFL168A.end();

Parameters

Nothing

Returns

Nothing

5.3.4 getOnewireID Method

bool getOnewireID(byte (&ID)[7]);

Description

DFL168A.getOnewireID(byte (&ID)[7]) will get ID from One wire bus device (iButton). It will return true if ID succeed to get, otherwise return false.

Syntax

DFL168A.getOnewireID(ID[7]);

Parameters

ID: the first parameter, 7 bytes array type, this is output parameter. The 7 bytes' ID will be put into this parameter.

Returns

bool

5.3.5 getDIN Method

bool getDIN(int portNo);

Description

getDIN(int portNo) will return Digital input of DFL168A. True means Logic High, false means Logic

Low.

Syntax

DFL168A.getDIN(portNo);

Parameters

portNo: the first parameter, int type, this is input parameter. It is DFL168A's Digital input number.

Returns

bool

5.3.6 setDOUT Method

void setDOUT(int portNo, bool Value);

Description

setDOUT(int portNo, bool Value) will set Digital output of DFL168A.

Syntax

DFL168A.setDOUT(portNo,Value);

Parameters

portNo: the first parameter, int type, this is input parameter. It is DFL168A's Digital output number.

Value: the second parameter, bool type, this is input parameter,true means Logic High, false means Logic Low.

Returns

Nothing

5.3.7 getAnalog Method

float getAnalog();

Description

getAnalog() will read analog input of DFL168A. value range: 0.0 to 999.00

Syntax

DFL168A.getAnalog();

Parameters

Nothing

Returns

float

5.3.8 beginTransparentSerial Method

void beginTransparentSerial();

Description

beginTransparentSerial() will use hardware Serial of Arduino to access DEV1 (Another serial port) of DFL168A directly. This hardware Serial was used for DFL168A Command in the past, now it is changed. You will use hardware Serial of Arduino to send/receive data to/from DEV1 of DFL168A directly. In general, DEV1 connects GPS module.

Syntax

DFL168A.beginTransparentSerial();

Parameters

Nothing

Returns

Nothing

5.3.9 endTransparentSerial Method

void endTransparentSerial();

Description

endTransparentSerial() will resume hardware Serial of Arduino to access DFL168A Command. For DEV1, now you cannot directly control it by hardware Serial of Arduino.

Syntax

DFL168A.endTransparentSerial();

Parameters

Nothing

Returns

Nothing

5.3.10 setExitTransparentKey Method

bool setExitTransparentKey(byte EndTransparentChar);

Description

setExitTransparentKey() will set up character for exiting Transparent Serial mode. When hardware Serial of DFL168A, which was used by DFL168A interface command, received this ASCII code from Arduino, DFL168A will resume the hardware Serial to use for DFL168A interface Command. This method return true when setting succeed. This method can't execute before running begin method.

Syntax

DFL168A.setExitTransparentKey(EndTransparentChar);

Parameters

EndTransparentChar: the first parameter, byte type, this is input parameter. It is ASCII code of exiting Transparent Serial.

Returns

bool

5.3.11 serialPortAvailable Method

bool serialPortAvailable();

Description

serialPortAvailable() will tell us whether hardware Serial of Arduino, which is used by DFL168A, can be used for access DEV1 directly. True means it can. Usually, after you run DFL168A. beginTransparentSerial(), this method will return true.

Syntax

DFL168A.serialPortAvailable();

Parameters

Nothing

Returns
bool

5.3.12 setSleepDelay Method

bool setSleepDelay(unsigned int SleepDelaySeconds);

Description

setSleepDelay(unsigned int SleepDelaySeconds) will set up sleep delay time in seconds. When vehicle data bus has no any activity for specified protocol, DFL168A IC will enter sleep state after this delay time if vehicle data bus still has no activity. The maximum delay time is 65535 seconds, which is about 18.2 hours. So if you want to disable sleep function, you should call this method once every 18.2 or less hours. This method will return true if call success, otherwise return false.

Syntax

DFL168A.setSleepDelay(unsigned int SleepDelaySeconds);

Parameters

SleepDelayms: the first parameter, unsigned int type, this is input parameter. It is delay time to sleep in seconds.

Returns
bool.

5.4 Inner Class J1939

5.4.1 Members

Class J1939 has the following members:

PGN65267;

PGN65262;

PGN65256;

PGN65269;

PGN65257;

PGN61444;

PGN61443;

PGN65270;

PGN65271;

PGN65272;

PGN65266;

PGN65263;

PGN65253;

PGN65214;

PGN65248;

PGN65276;

PGN65265;

PGN57344;

PGN64996;

PGN61445;

PGN65268;

Above members are actually object. We will explain these inner object later.

5.4.2 Methods

bool getVIN(String &VIN);
bool getDTC(byte &DTC_Num, long (&SPN)[5], byte (&FMI)[5], byte (&CM)[5],byte (&OC)[5], byte DTCFormat=1);
bool clearDTC();

5.4.2.1 getVIN Method

bool getVIN(String &VIN);

Description

J1939.getVIN(String &VIN) will get 19 characters' VIN number from vehicle. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.getVIN(VIN);

Parameters

VIN: the first parameter, String type, this is output parameter, It is VIN string of vehicle.

Returns
bool

5.4.2.2 getDTC Method

bool getDTC(byte &DTC_Num, long (&SPN)[5], byte (&FMI)[5], byte (&CM)[5],byte (&OC)[5], byte DTCFormat=1);

Description

This method will get DTC information of vehicle. It will return true if success, otherwise return false. Note: It only can get maximum of 5 DTC because of hardware serial buffer limit

Syntax

DFL168A.J1939.getDTC(DTC_Num,SPN,FMI,CM,OC,DTCFormat);

Parameters

DTC_Num: the first parameter, byte type, this is output parameter, It is quantity of vehicle DTC.

SPN: the second parameter, 5 elements' long array type, this is output parameter, It is SPN number of vehicle.

FMI: the third parameter, 5 elements' byte array type,this is output parameter,It is FMI of vehicle.

OC: the third parameter, 5 elements' byte array type,this is output parameter, It is OC of vehicle.

DTCFormat: the fourth parameter, byte type, this is input parameter,It is DTC Format Version of vehicle. It can be 1, 2, 3, and 4

Returns
bool

5.4.2.3 clearDTC Method

bool clearDTC();

Description

J1939.clearDTC() will clear DTC of vehicle. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.clearDTC();

Parameters

Nothing

Returns
bool

5.4.3 Inner Class PGN65267

5.4.3.1 Methods

bool refresh();

bool getLatitude(float &Latitude);

bool getLongitude(float &Longitude);

5.4.3.1.1 refresh Method

bool refresh();

Description

PGN65267.refresh() will refresh PGN65267 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65267.refresh();

Parameters

Nothing

Returns
bool

5.4.3.1.2 getLatitude Method

bool getLatitude(float &Latitude);

Description

PGN65267.getLatitude(float &Latitude) will get Latitude from vehicle. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65267.getLatitude(Latitude);

Parameters

Latitude: the first parameter, float type, this is output parameter, It is latitude of vehicle location.

Returns
bool

Example:

```
#include <DFL168A.h>

DFL168A myDFL168A(&Serial1, J1939_PROTOCOL, 1000,250000, 9600,500);
bool Vehicle_OK;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(57600);
  pinMode(13,OUTPUT);
  digitalWrite(13,LOW);
  Vehicle_OK=myDFL168A.begin();
  if (Vehicle_OK) digitalWrite(13,HIGH); else Serial.println("Vehicle is not
```

```
    ready!");
}

void loop() {
    // put your main code here, to run repeatedly:
    float Latitude;
    float Longitude;
    if (Vehicle_OK)
    {
        if (myDFL168A.J1939.PGN65267.refresh())
        {
            Serial.println("Success-PGN65267 Refresh");
            if (myDFL168A.J1939.PGN65267.getLatitude(Latitude))
            {
                Serial.print("Latitude: ");
                Serial.print(Latitude); Serial.println(" Degree");
            }
            else
            {
                Serial.println("Fail to get Latitude");
            }
            if (myDFL168A.J1939.PGN65267.getLongitude(Longitude))
            {
                Serial.print("Longitude: ");
                Serial.print(Longitude); Serial.println(" Degree");
            }
            else
            {
                Serial.println("Fail to get longitude");
            }
        }
        else
        {
            Serial.println("Fail to Refresh in PGN65267");
        }
    }
    delay(3000);

}
```

5.4.3.1.3 getLongitude Method

bool getLongitude(float &Longitude);

Description

PGN65267.getLongitude(float &Longitude) will get Longitude from vehicle. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65267.getLongitude(Longitude);

Parameters

Longitude: the first parameter, float type, this is output parameter, It is Longitude of vehicle location.

Returns
bool

5.4.4 Inner Class PGN65262

5.4.4.1 Methods

bool refresh();
bool getCoolantTemperature(int &temp);
bool getFuelTemp(int &temp);
bool getOilTemp(int &temp);

5.4.4.1.1 refresh Method

bool refresh();

Description

PGN65262.refresh() will refresh PGN65262 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65262.refresh();

Parameters

Nothing

Returns
bool

5.4.4.1.2 getCoolantTemperature Method

bool getCoolantTemperature(int &temp);

Description

PGN65262.getCoolantTemperature(int &temp) will get engine coolant temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65262.getCoolantTemperature(temp);

Parameters

temp: the first parameter, int type, this is output parameter, It is engine coolant temperature in Celsius degree.

Returns
bool

5.4.4.1.3 getFuelTemp Method

bool getFuelTemp(int &temp);

Description

PGN65262.getFuelTemp(int &temp) will get fuel temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65262.getFuelTemp(temp);

Parameters

temp: the first parameter, int type, this is output parameter, It is fuel temperature in Celsius degree.

Returns
bool

5.4.4.1.4 getOilTemp Method

bool getOilTemp(int &temp);

Description

PGN65262.getOilTemp(int &temp) will get engine oil temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65262.getOilTemp(temp);

Parameters

temp: the first parameter, int type, this is output parameter, It is engine oil temperature in Celsius degree.

Returns
bool

5.4.5 Inner Class PGN65256

5.4.5.1 Methods

bool refresh();

bool getAltitude(float &Altitude);

bool getNavBasedSpeed(float &Speed);

5.4.5.1.1 refresh Method

bool refresh();

Description

PGN65256.refresh() will refresh PGN65256 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65256.refresh();

Parameters

Nothing

Returns
bool

5.4.5.1.2 getAltitude Method

bool getAltitude(float &Altitude);

Description

PGN65256.getAltitude(float &Altitude) will get Altitude from vehicle. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65256.getAltitude(Altitude);

Parameters

Altitude: the first parameter, float type, this is output parameter, It is Altitude of vehicle location.

Returns

bool

5.4.5.1.3 getNavBasedSpeed Method

bool getNavBasedSpeed(float &Speed);

Description

PGN65256.getNavBasedSpeed(float &Speed) will get vehicle speed based on navigation in Km/h. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65256.getNavBasedSpeed(Speed);

Parameters

Speed: the first parameter, float type, this is output parameter, It is vehicle speed based on navigation in Km/h

Returns
bool

5.4.6 Inner Class PGN65269

5.4.6.1 Methods

bool refresh();

bool getBarometricPressure(float &BaroPressure);

bool getAmbientTemp(int &AmbientTemp);

bool getInletTemp(int &InletTemp);

bool getRoadTemp(int &RoadTemp);

bool getCabInteriorTemp(int &CabInteriorTemp);

5.4.6.1.1 refresh Method

bool refresh();

Description

PGN65269.refresh() will refresh PGN65269 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65269.refresh();

Parameters

Nothing

Returns
bool

5.4.6.1.2 getBarometricPressure Method

bool getBarometricPressure(float &BaroPressure);

Description

PGN65269.getBarometricPressure(float &BaroPressure) will get Barometric Pressure in KPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65269.getBarometricPressure(BaroPressure);

Parameters

BaroPressure: the first parameter, float type, this is output parameter, It is Barometric Pressure in kPa

Returns
bool

5.4.6.1.3 getAmbientTemp Method

bool getAmbientTemp(int &AmbientTemp);

Description

PGN65269.getAmbientTemp(int &AmbientTemp) will get Ambient Air Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65269.getAmbientTemp(AmbientTemp);

Parameters

AmbientTemp: the first parameter, int type, this is output parameter, It is Ambient Air Temperature in Celsius degree.

Returns
bool

5.4.6.1.4 getInletTemp Method

bool getInletTemp(int &InletTemp);

Description

PGN65269.getInletTemp(int &InletTemp) will get Engine Air Inlet Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65269.getInletTemp(InletTemp);

Parameters

InletTemp: the first parameter, int type, this is output parameter, It is Air Inlet Temperature in Celsius degree.

Returns

bool

5.4.6.1.5 getRoadTemp Method

bool getRoadTemp(int &RoadTemp);

Description

PGN65269.getRoadTemp(int &RoadTemp) will get road Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65269.getRoadTemp(RoadTemp);

Parameters

RoadTemp: the first parameter, int type, this is output parameter, It is road Temperature in Celsius degree.

Returns

bool

5.4.6.1.6 getCabInteriorTemp Method

bool getCabInteriorTemp(int &CabInteriorTemp);

Description

PGN65269.getCabInteriorTemp(int &CabInteriorTemp) will get Cab Interior Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65269.getCabInteriorTemp(CabInteriorTemp);

Parameters

CabInteriorTemp: the first parameter, int type, this is output parameter, It is Cab Interior Temperature in Celsius degree.

Returns

bool

5.4.7 Inner Class PGN65257

5.4.7.1 Methods

bool refresh();
bool getEngineTripFuel(float &EngineTripFuel);
bool getEngineTotalFuelUsed(float &EngineTotalFuelUsed);

5.4.7.1.1 refresh Method

bool refresh();

Description

PGN65257.refresh() will refresh PGN65257 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65257.refresh();

Parameters

Nothing

Returns
bool

5.4.7.1.2 getEngineTripFuel Method

bool getEngineTripFuel(float &EngineTripFuel);

Description

PGN65257.getEngineTripFuel(float &EngineTripFuel) will get Engine Trip Fuel in L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65257.getEngineTripFuel(EngineTripFuel);

Parameters

EngineTripFuel: the first parameter, float type, this is output parameter, It is Engine Trip Fuel in L.

Returns
bool

5.4.7.1.3 getEngineTotalFuelUsed Method

bool getEngineTotalFuelUsed(float &EngineTotalFuelUsed);

Description

PGN65257.getEngineTotalFuelUsed(float &EngineTotalFuelUsed) will get Engine Total Fuel Used in L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65257.getEngineTotalFuelUsed(EngineTotalFuelUsed);

Parameters

EngineTotalFuelUsed: the first parameter, float type, this is output parameter, It is Engine Total Fuel Used in L.

Returns
bool

5.4.8 Inner Class PGN61444

5.4.8.1 Methods

bool refresh();
bool getActualEngineTorque(int &ActualEngineTorque); //-125 to +125 (%)
bool getEngineSpeed(int &EngineSpeed);

5.4.8.1.1 refresh Method

bool refresh();

Description

PGN61444.refresh() will refresh PGN61444 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN61444.refresh();

Parameters

Nothing

Returns
bool

5.4.8.1.2 getActualEngineTorque Method

bool getActualEngineTorque(int &ActualEngineTorque);

Description

PGN61444.getActualEngineTorque(int &ActualEngineTorque) will get Actual Engine - Percent

Torque. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61444.getActualEngineTorque(ActualEngineTorque);

Parameters

ActualEngineTorque: the first parameter, int type, this is output parameter, It is Actual Engine - Percent Torque.

Returns
bool

5.4.8.1.3 getEngineSpeed Method

bool getEngineSpeed(int &EngineSpeed);

Description

PGN61444.getEngineSpeed(int &EngineSpeed) will get engine speed in rpm. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61444.getEngineSpeed(EngineSpeed);

Parameters

EngineSpeed: the first parameter, int type, this is output parameter, It is engine speed in rpm.

Returns
bool

5.4.9 Inner Class PGN61443

5.4.9.1 Methods

bool refresh();
bool getAccelPedalPosi1(float &AccelPedalPosi1);
bool getAccelPedalPosi2(float &AccelPedalPosi2);
bool getEnginePerLoadAtCurrSpeed(int &EnginePerLoadAtCurrSpeed);

5.4.9.1.1 refresh Method

bool refresh();

Description

PGN61443.refresh() will refresh PGN61443 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call

this method firstly.

Syntax

DFL168A.J1939.PGN61443.refresh();

Parameters

Nothing

Returns
bool

5.4.9.1.2 getAccelPedalPosi1 Method

bool getAccelPedalPosi1(float &AccelPedalPosi1);

Description

PGN61443.getAccelPedalPosi1(float &AccelPedalPosi1) will get Accelerator Pedal Position 1 in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61443.getAccelPedalPosi1(AccelPedalPosi1);

Parameters

AccelPedalPosi1: the first parameter, float type, this is output parameter, It is Accelerator Pedal Position 1 in percentage.

Returns
bool

5.4.9.1.3 getAccelPedalPosi2 Method

bool getAccelPedalPosi2(float &AccelPedalPosi2);

Description

PGN61443.getAccelPedalPosi2(float &AccelPedalPosi2) will get Accelerator Pedal Position 2 in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61443.getAccelPedalPosi2(AccelPedalPosi2);

Parameters

AccelPedalPosi2: the first parameter, float type, this is output parameter, It is Accelerator Pedal Position 2 in percentage.

Returns

bool

5.4.9.1.4 getEnginePerLoadAtCurrSpeed Method

bool getEnginePerLoadAtCurrSpeed(int &EnginePerLoadAtCurrSpeed);

Description

PGN61443.getEnginePerLoadAtCurrSpeed(int &EnginePerLoadAtCurrSpeed) will get Engine Percent Load At Current Speed. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61443.getEnginePerLoadAtCurrSpeed(EnginePerLoadAtCurrSpeed);

Parameters

EnginePerLoadAtCurrSpeed: the first parameter, int type, this is output parameter, It is Engine Percent Load At Current Speed.

Returns

bool

5.4.10 Inner Class PGN65270

5.4.10.1 Methods

bool refresh();
bool getIntakeManifoldPressure(int &IntakeManifoldPressure);
bool getIntakeManifoldTemp(int &IntakeManifoldTemp);
bool getEngineAirInletPressure(int &EngineAirInletPressure);
bool getEngineExhaustGasTemp(int &EngineExhaustGasTemp);
bool getEngineAirFilterDiffPressure(float &EngineAirFilterDiffPressure);

5.4.10.1.1 refresh Method

bool refresh();

Description

PGN65270.refresh() will refresh PGN65270 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65270.refresh();

Parameters

Nothing

Returns
bool

5.4.10.1.2 getIntakeManifoldPressure Method

bool getIntakeManifoldPressure(int &IntakeManifoldPressure);

Description

PGN65270.getIntakeManifoldPressure(int &IntakeManifoldPressure) will get Engine Intake Manifold #1 Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65270.getIntakeManifoldPressure(IntakeManifoldPressure);

Parameters

IntakeManifoldPressure: the first parameter, int type, this is output parameter, It is Engine Intake Manifold #1 Pressure in kPa.

Returns
bool

5.4.10.1.3 getIntakeManifoldTemp Method

bool getIntakeManifoldTemp(int &IntakeManifoldTemp);

Description

PGN65270.getIntakeManifoldTemp(int &IntakeManifoldTemp) will get Engine Intake Manifold 1 Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65270.getIntakeManifoldTemp(IntakeManifoldTemp);

Parameters

IntakeManifoldTemp: the first parameter, int type, this is output parameter, It is Engine Intake Manifold 1 Temperature in Celsius degree.

Returns
bool

5.4.10.1.4 getEngineAirInletPressure Method

bool getEngineAirInletPressure(int &EngineAirInletPressure);

Description

PGN65270.getEngineAirInletPressure(int &EngineAirInletPressure) will get Engine Air Inlet Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65270.getEngineAirInletPressure(EngineAirInletPressure);

Parameters

EngineAirInletPressure: the first parameter, int type, this is output parameter, It is Engine Air Inlet Pressure in kPa.

Returns
bool

5.4.10.1.5 getEngineExhaustGasTemp Method

bool getEngineExhaustGasTemp(int &EngineExhaustGasTemp);

Description

PGN65270.getEngineExhaustGasTemp(int &EngineExhaustGasTemp) will get Engine Exhaust Gas Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65270.getEngineExhaustGasTemp(EngineExhaustGasTemp);

Parameters

EngineExhaustGasTemp: the first parameter, int type, this is output parameter, It is Engine Exhaust Gas Temperature in Celsius degree.

Returns
bool

5.4.10.1.6 getEngineAirFilterDiffPressure Method

bool getEngineAirFilterDiffPressure(float &EngineAirFilterDiffPressure);

Description

PGN65270.getEngineAirFilterDiffPressure(float &EngineAirFilterDiffPressure) will get Engine Air Filter 1 Differential Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65270.getEngineAirFilterDiffPressure(EngineAirFilterDiffPressure);

Parameters

EngineAirFilterDiffPressure: the first parameter, float type, this is output parameter, It is Engine Air Filter 1 Differential Pressure in kPa.

Returns

bool

5.4.11 Inner Class PGN65271

5.4.11.1 Methods

bool refresh();
bool getAlternatorVoltage(float &AlternatorVoltage);
bool getElectricalVoltage(float &ElectricalVoltage);
bool getBatteryVoltage(float &BatteryVoltage);

5.4.11.1.1 refresh Method

bool refresh();

Description

PGN65271.refresh() will refresh PGN65271 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65271.refresh();

Parameters

Nothing

Returns
bool

5.4.11.1.2 getAlternatorVoltage Method

bool getAlternatorVoltage(float &AlternatorVoltage);

Description

PGN65271.getAlternatorVoltage(float &AlternatorVoltage) will get Charging System Potential (Voltage). It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65271.getAlternatorVoltage(AlternatorVoltage);

Parameters

AlternatorVoltage: the first parameter, float type, this is output parameter, It is Charging System Potential (Voltage).

Returns
bool

5.4.11.1.3 getElectricalVoltage Method

bool getElectricalVoltage(float &ElectricalVoltage);

Description

PGN65271.getElectricalVoltage(float &ElectricalVoltage) will get Battery Potential / Power Input 1. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65271.getElectricalVoltage(ElectricalVoltage);

Parameters

ElectricalVoltage: the first parameter, float type, this is output parameter, It is Battery Potential / Power Input 1.

Returns
bool

5.4.11.1.4 getBatteryVoltage Method

bool getBatteryVoltage(float &BatteryVoltage);

Description

PGN65271.getBatteryVoltage(float &BatteryVoltage) will get Keyswitch Battery Potential. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65271.getBatteryVoltage(BatteryVoltage);

Parameters

BatteryVoltage: the first parameter, float type, this is output parameter, It is Keyswitch Battery Potential.

Returns
bool

5.4.12 Inner Class PGN65272

5.4.12.1 Methods

bool refresh();
bool getTransmissionOilLevel(float &Percent);
bool getTransmissionOilLevelHighLow(float &HighLow);
bool getTransmissionOilPressure(float &Pressure);
bool getTransmissionOilTemp(float &Temperature);.

5.4.12.1.1 refresh Method

bool refresh();

Description

PGN65272.refresh() will refresh PGN65272 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65272.refresh();

Parameters

Nothing

Returns
bool

5.4.12.1.2 getTransmissionOilLevel Method

bool getTransmissionOilLevel(float &Percent);

Description

PGN65272.getTransmissionOilLevel(float &Percent) will get Transmission Oil Level in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65272.getTransmissionOilLevel(Percent);

Parameters

Percent: the first parameter, float type, this is output parameter, It is Transmission Oil Level in percentage.

Returns
bool

5.4.12.1.3 getTransmissionOilLevelHighLow Method

bool getTransmissionOilLevelHighLow(float &HighLow);

Description

PGN65272.getTransmissionOilLevelHighLow(float &HighLow) will get Amount of current volume of transmission sump oil compared to recommended volume. Positive values indicate overfill. Zero means the transmission fluild is filled to the recommended level. Unit is L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65272.getTransmissionOilLevelHighLow(HighLow);

Parameters

HighLow: the first parameter, float type, this is output parameter, It is Amount of current volume of transmission sump oil compared to recommended volume. Unit is L.

Returns
bool

5.4.12.1.4 getTransmissionOilPressure Method

bool getTransmissionOilPressure(float &Pressure);

Description

PGN65272.getTransmissionOilPressure(float &Pressure) will get Transmission Oil Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65272.getTransmissionOilPressure(Pressure);

Parameters

Pressure: the first parameter, float type, this is output parameter, It is Transmission Oil Pressure in kPa.

Returns
bool

5.4.12.1.5 getTransmissionOilTemp Method

bool getTransmissionOilTemp(float &Temperature);

Description

PGN65272.getTransmissionOilTemp(float &Temperature) will get Transmission Oil Temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65272.getTransmissionOilTemp(Temperature);

Parameters

Temperature: the first parameter, float type, this is output parameter, It is Transmission Oil Temperature in Celsius degree.

Returns
bool

5.4.13 Inner Class PGN65266

5.4.13.1 Methods

bool refresh();
bool getFuelRate(float &FuelRate);
bool getInstantFuelEconomy(float &InstantFuelEconomy);
bool getAvgFuelEconomy(float &AvgFuelEconomy);
bool getEngineThrottlePos(float &EngineThrottlePos);

5.4.13.1.1 refresh Method

bool refresh();

Description

PGN65266.refresh() will refresh PGN65266 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65266.refresh();

Parameters

Nothing

Returns
bool

5.4.13.1.2 getFuelRate Method

bool getFuelRate(float &FuelRate);

Description

PGN65266.getFuelRate(float &FuelRate) will get Engine Fuel Rate in L/H. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65266.getFuelRate(FuelRate);

Parameters

FuelRate: the first parameter, float type, this is output parameter, It is Engine Fuel Rate in L/H.

Returns
bool

5.4.13.1.3 getInstantFuelEconomy Method

bool getInstantFuelEconomy(float &InstantFuelEconomy);

Description

PGN65266.getInstantFuelEconomy(float &InstantFuelEconomy) will get Engine Instantaneous Fuel Economy in Km/L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65266.getInstantFuelEconomy(InstantFuelEconomy);

Parameters

InstantFuelEconomy: the first parameter, float type, this is output parameter, It is Engine Instantaneous Fuel Economy in Km/L.

Returns
bool

5.4.13.1.4 getAvgFuelEconomy Method

bool getAvgFuelEconomy(float &AvgFuelEconomy);

Description

PGN65266.getAvgFuelEconomy(float &AvgFuelEconomy) will get Engine Average Fuel Economy in Km/L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65266.getAvgFuelEconomy(AvgFuelEconomy);

Parameters

AvgFuelEconomy: the first parameter, float type, this is output parameter, It is Engine Average Fuel Economy in Km/L.

Returns
bool

5.4.13.1.5 getEngineThrottlePos Method

bool getEngineThrottlePos(float &EngineThrottlePos);

Description

PGN65266.getEngineThrottlePos(float &EngineThrottlePos) will get Engine Throttle Position in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65266.getEngineThrottlePos(EngineThrottlePos);

Parameters

EngineThrottlePos: the first parameter, float type, this is output parameter, It is Engine Throttle Position in percentage.

Returns
bool

5.4.14 Inner Class PGN65263

5.4.14.1 Methods

bool refresh();
bool getFueDeliveryPressure(int &FueDeliveryPressure);
bool getEngineOilLevel(float &EngineOilLevel);
bool getEngineOilPressure(int &EngineOilPressure);
bool getEngineCoolantPressure(int &EngineCoolantPressure);
bool getEngineCoolantLevel(float &EngineCoolantLevel);

5.4.14.1.1 refresh Method

bool refresh();

Description

PGN65263.refresh() will refresh PGN65263 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65263.refresh();

Parameters

Nothing

Returns
bool

5.4.14.1.2 getFueDeliveryPressure Method

bool getFueDeliveryPressure(int &FueDeliveryPressure);

Description

PGN65263.getFueDeliveryPressure(int &FueDeliveryPressure) will get Engine Fuel Delivery Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65263.getFueDeliveryPressure(FueDeliveryPressure);

Parameters

FueDeliveryPressure: the first parameter, int type, this is output parameter, It is Engine Fuel Delivery Pressure in kPa.

Returns
bool

5.4.14.1.3 getEngineOilLevel Method

```
bool getEngineOilLevel(float &EngineOilLevel);
```

Description

PGN65263.getEngineOilLevel(float &EngineOilLevel) will get Engine Oil Level in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65263.getEngineOilLevel(EngineOilLevel);

Parameters

EngineOilLevel: the first parameter, float type, this is output parameter, It is Engine Oil Level in percentage

Returns
bool

5.4.14.1.4 getEngineOilPressure Method

bool getEngineOilPressure(int &EngineOilPressure);

Description

PGN65263.getEngineOilPressure(int &EngineOilPressure) will get Engine Oil Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65263.getEngineOilPressure(EngineOilPressure);

Parameters

EngineOilPressure: the first parameter, int type, this is output parameter, It is Engine Oil Pressure in kPa.

Returns
bool

5.4.14.1.5 getEngineCoolantPressure Method

bool getEngineCoolantPressure(int &EngineCoolantPressure);

Description

PGN65263.getEngineCoolantPressure(int &EngineCoolantPressure) will get Engine Coolant Pressure in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65263.getEngineCoolantPressure(EngineCoolantPressure);

Parameters

EngineCoolantPressure: the first parameter, int type, this is output parameter, It is Engine Coolant Pressure in kPa.

Returns
bool

5.4.14.1.6 getEngineCoolantLevel Method

bool getEngineCoolantLevel(float &EngineCoolantLevel);

Description

PGN65263.getEngineCoolantLevel(float &EngineCoolantLevel) will get Engine Coolant Level in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65263.getEngineCoolantLevel(EngineCoolantLevel);

Parameters

EngineCoolantLevel: the first parameter, float type, this is output parameter, It is Engine Coolant Level in percentage

Returns
bool

5.4.15 Inner Class PGN65253

5.4.15.1 Methods

bool refresh();
bool getTotalEngineHours(float &TotalEngineHours);
bool getTotalEngineRevolutions(float &TotalEngineRevolutions);

5.4.15.1.1 refresh Method

bool refresh();

Description

PGN65253.refresh() will refresh PGN65253 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65253.refresh();

Parameters

Nothing

Returns

bool

5.4.15.1.2 getTotalEngineHours Method

bool getTotalEngineHours(float &TotalEngineHours);

Description

PGN65253.getTotalEngineHours(float &TotalEngineHours) will get Engine Total Hours of Operation. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65253.getTotalEngineHours(TotalEngineHours);

Parameters

TotalEngineHours: the first parameter, float type, this is output parameter, It is Engine Total Hours of Operation.

Returns

bool

5.4.15.1.3 getTotalEngineRevolutions Method

bool getTotalEngineRevolutions(float &TotalEngineRevolutions);

Description

PGN65253.getTotalEngineRevolutions(float &TotalEngineRevolutions) will get Engine Total Revolutions. Unit is r. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65253.getTotalEngineRevolutions(TotalEngineRevolutions);

Parameters

TotalEngineRevolutions: the first parameter, float type, this is output parameter, It is Engine Total Revolutions. Unit is r.

Returns
bool

5.4.16 Inner Class PGN65214

5.4.16.1 Methods

bool refresh();
bool getRatedEngineSpeed(float &RatedEngineSpeed);

5.4.16.1.1 refresh Method

bool refresh();

Description

PGN65214.refresh() will refresh PGN65214 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65214.refresh();

Parameters

Nothing

Returns
bool

5.4.16.1.2 getRatedEngineSpeed Method

bool getRatedEngineSpeed(float &RatedEngineSpeed);

Description

PGN65214.getRatedEngineSpeed(float &RatedEngineSpeed) will get Engine Rated Speed in rpm. Unit is r. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65214.getRatedEngineSpeed(RatedEngineSpeed);

Parameters

RatedEngineSpeed: the first parameter, float type, this is output parameter, It is Engine Rated Speed

in rpm.

Returns
bool

5.4.17 Inner Class PGN65248

5.4.17.1 Methods

bool refresh();
bool getTripDistance(float &TripDistance);
bool getTotalDistance(float &TotalDistance);

5.4.17.1.1 refresh Method

bool refresh();

Description

PGN65248.refresh() will refresh PGN65248 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65248.refresh();

Parameters

Nothing

Returns
bool

5.4.17.1.2 getTripDistance Method

bool getTripDistance(float &TripDistance);

Description

PGN65248.getTripDistance(float &TripDistance) will get Trip Distance in Km. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65248.getTripDistance(TripDistance);

Parameters

TripDistance: the first parameter, float type, this is output parameter, It is Trip Distance in Km.

Returns
bool

5.4.17.1.3 getTotalDistance Method

bool getTotalDistance(float &TotalDistance);

Description

PGN65248.getTotalDistance(float &TotalDistance) will get Total Vehicle Distance in Km. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65248.getTotalDistance(TotalDistance);

Parameters

TotalDistance: the first parameter, float type, this is output parameter, It is Total Vehicle Distance in Km.

Returns
bool

5.4.18 Inner Class PGN65276

5.4.18.1 Methods

bool refresh();
bool getWasherFluidLevel(float &WasherFluidLevel);
bool getFuelLevel1(float &FuelLevel1);
bool getFuelLevel2(float &FuelLevel2);

5.4.18.1.1 refresh Method

bool refresh();

Description

PGN65276.refresh() will refresh PGN65276 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65276.refresh();

Parameters

Nothing

Returns
bool

5.4.18.1.2 getWasherFluidLevel Method

bool getWasherFluidLevel(float &WasherFluidLevel);

Description

PGN65276.getWasherFluidLevel(float &WasherFluidLevel) will get Washer Fluid Level in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65276.getWasherFluidLevel(WasherFluidLevel);

Parameters

WasherFluidLevel: the first parameter, float type, this is output parameter, It is Washer Fluid Level in percentage

Returns
bool

5.4.18.1.3 getFuelLevel1 Method

bool getFuelLevel1(float &FuelLevel1);

Description

PGN65276.getFuelLevel1(float &FuelLevel1) will get Fuel Level 1 in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65276.getFuelLevel1(FuelLevel1);

Parameters

FuelLevel1: the first parameter, float type, this is output parameter, It is Fuel Level 1 in percentage

Returns
bool

5.4.18.1.4 getFuelLevel2 Method

bool getFuelLevel2(float &FuelLevel2);

Description

PGN65276.getFuelLevel2(float &FuelLevel2) will get Fuel Level 2 in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65276.getFuelLevel2(FuelLevel2);

Parameters

FuelLevel2: the first parameter, float type, this is output parameter, It is Fuel Level 2 in percentage

Returns
bool

5.4.19 Inner Class PGN65265

5.4.19.1 Methods

bool refresh();
bool getWheelBasedVehicleSpeed(float &WheelBasedVehicleSpeed);
bool getParkingBrake(bool &ParkingBrakeSet);
bool getBrake(bool &BrakePedalDepressed);

5.4.19.1.1 refresh Method

bool refresh();

Description

PGN65265.refresh() will refresh PGN65265 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65265.refresh();

Parameters

Nothing

Returns
bool

5.4.19.1.2 getWheelBasedVehicleSpeed Method

bool getWheelBasedVehicleSpeed(float &WheelBasedVehicleSpeed);

Description

PGN65265.getWheelBasedVehicleSpeed(float &WheelBasedVehicleSpeed) will get Wheel-Based Vehicle Speed in km/h. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65265.getWheelBasedVehicleSpeed(WheelBasedVehicleSpeed);

Parameters

WheelBasedVehicleSpeed: the first parameter, float type, this is output parameter, It is Wheel-

Based Vehicle Speed in km/h.

Returns
bool

5.4.19.1.3 getParkingBrake(bool &ParkingBrakeSet)

bool getParkingBrake(bool &ParkingBrakeSet);

Description

PGN65265.getParkingBrake(bool &ParkingBrakeSet) will get Parking brake switch status: True/False. Status "True" means Parking brake set. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65265.getParkingBrake(ParkingBrakeSet);

Parameters

ParkingBrakeSet: the first parameter, bool type, this is output parameter, It denotes whether Parking brake switch set.

Returns
bool

5.4.19.1.4 getBrake(bool &BrakePedalDepressed)

bool getBrake(bool &BrakePedalDepressed);

Description

PGN65265.getBrake(bool &BrakePedalDepressed) will get Brake switch status: True/False. Status "True" means Brake pedal depressed. Status "False" means Brake pedal released. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.getBrake(BrakePedalDepressed);

Parameters

BrakePedalDepressed: the first parameter, bool type, this is output parameter, It denotes whether Brake pedal depressed.

Returns
bool

5.4.20 Inner Class PGN57344

5.4.20.1 Methods

bool refresh();
bool getSeatBelt(bool buckled);

5.4.20.1.1 refresh Method

bool refresh();

Description

PGN57344.refresh() will refresh PGN57344 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN57344.refresh();

Parameters

Nothing

Returns
bool

5.4.20.1.2 getSeatBelt Method

bool getSeatBelt(bool &buckled);

Description

PGN57344.getSeatBelt(bool &buckled) will get status of Seat Belt Switch. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN57344.getSeatBelt(buckled);

Parameters

buckled: the first parameter, bool type, this is output parameter, true means that Seat Belt is buckled, false means that Seat Belt is not buckled

Returns
bool

5.4.21 Inner Class PGN64996

5.4.21.1 Methods

bool refresh();
bool getPayLoad(int &PayLoad);

5.4.21.1.1 refresh Method

bool refresh();

Description

PGN64996.refresh() will refresh PGN64996 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN64996.refresh();

Parameters

Nothing

Returns
bool

5.4.21.1.2 getSeatBelt Method

bool getPayLoad(int &PayLoad);

Description

PGN64996.getPayLoad(int &PayLoad) will get Payload Percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN64996.getPayLoad(PayLoad);

Parameters

PayLoad: the first parameter, int type, this is output parameter, 0 to 250 (%). The current payload of the equipment, reported as a percentage of the equipment's rated payload limit

Returns
bool

5.4.22 Inner Class PGN61445

5.4.22.1 Methods

bool refresh();
bool getCurrentGear(int &CurrentGear);
bool getSelectedGear(int &SelectedGear);

5.4.22.1.1 refresh Method

bool refresh();

Description

PGN61445.refresh() will refresh PGN61445 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN61445.refresh();

Parameters

Nothing

Returns

bool

5.4.22.1.2 getCurrentGear

bool getCurrentGear(int &CurrentGear);

Description

PGN61445.getCurrentGear(int &CurrentGear) will get Transmission Current Gear. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61445.getCurrentGear(CurrentGear);

Parameters

CurrentGear: the first parameter, int type, this is output parameter, -125 to 125(%) and 251%.
Negative values are reverse gears, positive values are forward gears, zero is neutral. 251 is park

Returns

bool

5.4.22.1.3 getSelectedGear

bool getSelectedGear(int &SelectedGear);

Description

PGN61445.getSelectedGear(int &SelectedGear) will get Transmission Selected Gear. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN61445.getSelectedGear(SelectedGear);

Parameters

SelectedGear: the first parameter, int type, this is output parameter, -125 to 125(%) and 251%.
Negative values are reverse gears, positive values are forward gears, zero is neutral. 251 is park

Returns
bool

5.4.23 Inner Class PGN65268

5.4.23.1 Methods

bool refresh();
bool getTirePressure(int &TirePressure);
bool getTireTemperature(float &Temperature);
bool getTireLocation(int &Front2RearNumber, int &Left2RighNumber);
bool getTireValvePressureMonitor(int &TireValvePressureMonitor);

5.4.23.1.1 refresh Method

bool refresh();

Description

PGN65268.refresh() will refresh PGN65268 data from vehicle. It will return true if success, otherwise return false. If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN61445.refresh();

Parameters

Nothing

Returns
bool

5.4.23.1.2 getTirePressure

bool getTirePressure(int &TirePressure);

Description

PGN65268.getTirePressure(int &TirePressure) will get tire pressure in kPa unit. It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65268.getTirePressure(TirePressure);

Parameters

TirePressure: the first parameter, int type, this is output parameter, Data rang is 0 to 1000 KPa.

Returns
bool

5.4.23.1.3 getTireTemperature

bool getTireTemperature(float &Temperature);

Description

PGN65268.getTireTemperature(float &Temperature) will get tire temperature in deg C unit . It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65268.getTireTemperature(Temperature);

Parameters

Temperature: the first parameter, float type, this is output parameter, Data range is -273 to 1734.96875 deg C

Returns
bool

5.4.23.1.4 getTireLocation

bool getTireLocation(int &Front2RearNumber, int &Left2RighNumber);

Description

PGN65268.getTireLocation(int &Front2RearNumber, int &Left2RighNumber) will identify which tire is associated with the parametric data in this PGN . It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65268.getTireLocation(Front2RearNumber, Left2RighNumber) ;

Parameters

Front2RearNumber: the first parameter, int type, this is output parameter, it represents a position number , counting front to rear on the vehicle.

Left2RighNumber: the second parameter, int type, this is output parameter, it represents a position number , counting left to right when facing in the direction of normal travel (forward).

Returns
bool

5.4.23.1.5 getTireValvePressureMonitor

```
bool getTireValvePressureMonitor(int &TireValvePressureMonitor);
```

Description

PGN65268.getTireValvePressureMonitor(int &TireValvePressureMonitor) will get the pressure level of tire . It will return true if success, otherwise return false.

Syntax

DFL168A.J1939.PGN65268.getTireValvePressureMonitor(TireValvePressureMonitor);

Parameters

TireValvePressureMonitor: the first parameter, int type, this is output parameter, Data range binary 000 to 110.

000 : Extreme over pressure. 001: Over pressure. 010 : No warning pressure. 011: Under pressure.

100: Extreme under pressure. 101: Not defined, 110: Error indictor.

Returns
bool

5.5 Inner Class J1708

5.5.1 Methods

bool getAirPressure(int &AirPressure);
bool getEngineOilPressure(int &EngineOilPressure);
bool getEngineCoolantPressure(int &EngineCoolantPressure);
bool getFuelLevel1(float &FuelLevel1);
bool getFuelLevel2(float &FuelLevel2);
bool getBarometricPressure(float &Pressure);
bool getEngineThrottlePos(float &EngineThrottlePos);
bool getWasherFluidLevel(float &WasherFluidLevel);
bool getVehicleSpeed(float &VehicleSpeed);
bool getAccelPedalPosi1(float &AccelPedalPosi1);
bool getAccelPedalPosi2(float &AccelPedalPosi2);
bool getAccelPedalPosi3(float &AccelPedalPosi3);
bool getEngineLoad(float &Percent);
bool getEngineOilLevel(float &EngineOilLevel);
bool getCoolantTemperature(int &temp);

bool getEngineCoolantLevel(float &EngineCoolantLevel);
bool getTransmissionOilLevel(float &Percent);
bool getTransmissionOilLevelHighLow(float &HighLow);
bool getTransmissionOilPressure(float &Pressure);
bool getTransmissionOilTemp(float &Temperature);
bool getPowerSpecificInstantFuelEconomy(float &Rate);
bool getAvgFuelRate(float &FuelRate);
bool getInstantFuelEconomy(float &InstantFuelEconomy);
bool getAvgFuelEconomy(float &AvgFuelEconomy);
bool getElectricalVoltage(float &BatteryVoltage);
bool getRatedEnginePower(float &Power);
bool getBatteryVoltage(float &BatteryVoltage);
bool getAlternatorVoltage(float &AlternatorVoltage);
bool getAmbientTemp(int &AmbientTemp);
bool getCargoAmbientTemp(int &CargoTemp);
bool getRoadTemp(int &RoadTemp);
bool getCabInteriorTemp(int &CabInteriorTemp);
bool getInletTemp(int &InletTemp);
bool getFuelTemp(int &temp);
bool getOilTemp(int &temp);
bool getCargoWeight(float &CargoW);
bool getEngineTripFuel(float &EngineTripFuel);
bool getEngineTotalFuelUsed(float &EngineTotalFuelUsed);
bool getFuelRate(float &FuelRate);
bool getRatedEngineSpeed(float &RatedEngineSpeed);
bool getEngineSpeed(int &EngineSpeed);
bool getIntakeManifoldTemp(int &IntakeManifoldTemp);
bool getPowerTakeoffStatus(bool &PTOModeActive, bool &ClutchSwitchOn, bool &BrakeSwitchOn, bool &AccelSwitchOn, bool &ResumeSwitchOn, bool &CoastSwitchOn, bool &SetSwitchOn, bool &PTOControlSwitchOn);
bool getTripDistance(float &TripDistance);
bool getTotalDistance(float &TotalDistance);
bool getTotalEngineHours(float &TotalEngineHours);
bool getTotalEngineRevolutions(float &TotalEngineRevolutions);
bool getVIN(String &VIN);
bool getDTC(byte &DTC_Num,byte &MID,int (&PID_SID)[8],bool (&IsPID)[8], byte (&FMI)[8],bool (&IsActive)[8],bool (&OccurrenceExist)[8],byte (&OccurrenceCount)[8]);
bool clearDTC(byte MID,int PID_SID, bool IsPID);
bool getFaultDescription(byte MID,int PID_SID, bool IsPID,byte FMI, String & FaultDescription);

bool getPIDSIDDescription(byte MID,int PID_SID, bool IsPID, String & PID_SID_Description);

5.5.1.1 getAirPressure Method

bool getAirPressure(int &AirPressure);

Description

getAirPressure(int &AirPressure) will get Gauge Pressure of air in system that utilizes compressed air to provide force between a lift axle and frame for purposes of
lifting or lowering the axle, unit is kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAirPressure(int AirPressure);

Parameters

AirPressure: the first parameter, int type, this is output parameter. It is Gauge Pressure of air in system in kPa.

Returns
bool

5.5.1.2 getEngineOilPressure Method

bool getEngineOilPressure(int &EngineOilPressure);

Description

getEngineOilPressure(int &EngineOilPressure) will get Gauge pressure of oil in the engine lubrication system as provided by the oil pump, unit is kPa .It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineOilPressure(int EngineOilPressure);

Parameters

EngineOilPressure: the first parameter, int type, this is output parameter. It is Gauge pressure of oil in the engine lubrication system in kPa.

Returns
bool

5.5.1.3 getEngineCoolantPressure Method

bool getEngineCoolantPressure(int &EngineCoolantPressure);

Description

getEngineCoolantPressure(int &EngineCoolantPressure) will get Gauge pressure of liquid found in the engine cooling system, unit is kPa .It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineCoolantPressure(int EngineCoolantPressure);

Parameters

EngineCoolantPressure: the first parameter, int type, this is output parameter. It is Gauge pressure of liquid found in the engine cooling system in kPa.

Returns
bool

5.5.1.4 getFuelLevel1 Method

bool getFuelLevel1(float &FuelLevel1);

Description

getFuelLevel1(float &FuelLevel1) will get ratio of volume of fuel to the total volume of the primary fuel storage container. Unit is percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getFuelLevel1(float FuelLevel1);

Parameters

FuelLevel1: the first parameter,float type, this is output parameter. It is ratio of volume of fuel to the total volume of the primary fuel storage container. Unit is percentage.

Returns
bool

5.5.1.5 getFuelLevel2 Method

bool getFuelLevel2(float &FuelLevel2);

Description

getFuelLevel2(float &FuelLevel2) will get ratio of volume of fuel to the total volume of the second fuel storage container. Unit is percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getFuelLevel2(float FuelLevel2);

Parameters

FuelLevel2: the first parameter,float type, this is output parameter. It is ratio of volume of fuel to the total volume of the second fuel storage container. Unit is percentage.

Returns

bool

5.5.1.6 getBarometricPressure Method

bool getBarometricPressure(float &Pressure);

Description

getBarometricPressure(float &Pressure) will get absolute air pressure of the atmosphere in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getBarometricPressure(float Pressure);

Parameters

Pressure: the first parameter,float type, this is output parameter. It is absolute air pressure of the atmosphere in kPa.

Returns

bool

5.5.1.7 getEngineThrottlePos Method

bool getEngineThrottlePos(float &EngineThrottlePos);

Description

getEngineThrottlePos(float &EngineThrottlePos) will get the position of the valve used to regulate the supply of a fluid, usually air or fuel/air mixture, to an engine. 0% represents no supply and 100% is full supply. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineThrottlePos(float EngineThrottlePos);

Parameters

EngineThrottlePos: the first parameter,float type, this is output parameter. It is the position of the valve used to regulate the supply of a fluid. This is percentage.

Returns

bool

5.5.1.8 getWasherFluidLevel Method

bool getWasherFluidLevel(float &WasherFluidLevel);

Description

getWasherFluidLevel(float &WasherFluidLevel) will get ratio of volume of liquid to total container volume of fluid reservoir in windshield wash system. It will return true if success, otherwise return

false.

Syntax

DFL168A.J1708.getWasherFluidLevel(float WasherFluidLevel);

Parameters

WasherFluidLevel: the first parameter,float type, this is output parameter. It is the ratio of volume of liquid to total container volume of fluid reservoir in windshield wash system. This is percentage.

Returns
bool

5.5.1.9 getVehicleSpeed Method

bool getVehicleSpeed(float &VehicleSpeed);

Description

getVehicleSpeed(float &VehicleSpeed) will get vehicle road speed in km/h. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getVehicleSpeed(float VehicleSpeed);

Parameters

WasherFluidLevel: the first parameter,float type, this is output parameter. It is vehicle road speed in km/h.

Returns
bool

5.5.1.10 getAccelPedalPosi1 Method

bool getAccelPedalPosi1(float &AccelPedalPosi1);

Description

getAccelPedalPosi1(float &AccelPedalPosi1) will get ratio of actual accelerator pedal position to maximum pedal position. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAccelPedalPosi1(float AccelPedalPosi1);

Parameters

AccelPedalPosi1: the first parameter,float type, this is output parameter. It is the ratio of actual accelerator pedal position to maximum pedal position. This is percentage.

Returns
bool

5.5.1.11 getAccelPedalPosi2 Method

bool getAccelPedalPosi2(float &AccelPedalPosi2);

Description

getAccelPedalPosi2(float &AccelPedalPosi2) will get ratio of actual accelerator pedal position to maximum pedal position. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAccelPedalPosi2(float AccelPedalPosi2);

Parameters

AccelPedalPosi2: the first parameter,float type, this is output parameter. It is the ratio of actual accelerator pedal position to maximum pedal position. This is percentage.

Returns
bool

5.5.1.12 getAccelPedalPosi3 Method

bool getAccelPedalPosi3(float &AccelPedalPosi3);

Description

getAccelPedalPosi3(float &AccelPedalPosi3) will get ratio of actual accelerator pedal position to maximum pedal position. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAccelPedalPosi3(float AccelPedalPosi3);

Parameters

AccelPedalPosi3: the first parameter,float type, this is output parameter. It is the ratio of actual accelerator pedal position to maximum pedal position. This is percentage.

Returns
bool

5.5.1.13 getEngineLoad Method

bool getEngineLoad(float &Percent);

Description

getEngineLoad(float &Percent) will get ratio of current output torque to maximum torque available at the current engine speed. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineLoad(float Percent);

Parameters

Percent: the first parameter,float type, this is output parameter. It is the ratio of current output torque to maximum torque available at the current engine speed. This is percentage.

Returns

bool

5.5.1.14 getEngineOilLevel Method

bool getEngineOilLevel(float &EngineOilLevel);

Description

getEngineOilLevel(float &EngineOilLevel) will get ratio of current volume of engine sump oil to maximum required volume. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineOilLevel(float EngineOilLevel);

Parameters

EngineOilLevel: the first parameter,float type, this is output parameter. It is the ratio of current volume of engine sump oil to maximum required volume. This is percentage.

Returns

bool

5.5.1.15 getCoolantTemperature Method

bool getCoolantTemperature(int &temp);

Description

getCoolantTemperature(int &temp) will get the temperature of liquid found in engine cooling system in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getCoolantTemperature(int temp);

Parameters

temp: the first parameter,int type, this is output parameter. It is the temperature of liquid found in engine cooling system in Celsius degree.

Returns

bool

5.5.1.16 getEngineCoolantLevel Method

bool getEngineCoolantLevel(float &EngineCoolantLevel);

Description

getEngineCoolantLevel(float &EngineCoolantLevel) will get ratio of volume of liquid found in engine cooling system to total cooling system volume. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineCoolantLevel(float EngineCoolantLevel);

Parameters

EngineCoolantLevel: the first parameter,float type, this is output parameter. It is the ratio of volume of liquid found in engine cooling system to total cooling system volume. This is percentage.

Returns

bool

5.5.1.17 getTransmissionOilLevel Method

bool getTransmissionOilLevel(float &Percent);

Description

getTransmissionOilLevel(float &Percent) will get ratio of volume of transmission sump oil to recommended volume. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTransmissionOilLevel(float Percent);

Parameters

Percent: the first parameter,float type, this is output parameter. It is the ratio of volume of transmission sump oil to recommended volume. This is percentage.

Returns

bool

5.5.1.18 getTransmissionOilLevelHighLow Method

bool getTransmissionOilLevelHighLow(float &HighLow);

Description

getTransmissionOilLevelHighLow(float &HighLow) will get amount of current volume of transmission sump oil compared to recommended volume. Unit is L. It will return true if success, otherwise return

false.

Syntax

DFL168A.J1708.getTransmissionOilLevelHighLow(float HighLow);

Parameters

HighLow: the first parameter,float type, this is output parameter. It is the amount of current volume of transmission sump oil compared to recommended volume. Unit is L.

Returns

bool

5.5.1.19 getTransmissionOilPressure Method

bool getTransmissionOilPressure(float &Pressure);

Description

getTransmissionOilPressure(float &Pressure) will get gage pressure of lubrication fluid in transmission, measured after pump in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTransmissionOilPressure(float Pressure);

Parameters

Pressure: the first parameter,float type, this is output parameter. It is gage pressure of lubrication fluid in transmission, measured after pump in kPa.

Returns

bool

5.5.1.20 getTransmissionOilTemp Method

bool getTransmissionOilTemp(float &Temperature);

Description

getTransmissionOilTemp(float &Temperature) will get temperature of transmission lubricant in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTransmissionOilTemp(float Temperature);

Parameters

Temperature: the first parameter,float type, this is output parameter. It is the temperature of transmission lubricant in Celsius degree.

Returns
bool

5.5.1.21 getPowerSpecificInstantFuelEconomy Method

bool getPowerSpecificInstantFuelEconomy(float &Rate);

Description

getPowerSpecificInstantFuelEconomy(float &Rate) will get instantaneous fuel economy of the engine, typically for off-highway equipment in kWh/L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getPowerSpecificInstantFuelEconomy(float Rate);

Parameters

Rate: the first parameter, float type, this is output parameter. It is the instantaneous fuel economy of the engine in kWh/L.

Returns
bool

5.5.1.22 getAvgFuelRate Method

bool getAvgFuelRate(float &FuelRate);

Description

getAvgFuelRate(float &FuelRate) will get continuous averaging fuel per hour per segment of engine operation in L/s. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAvgFuelRate(float FuelRate);

Parameters

FuelRate: the first parameter, float type, this is output parameter. It is the continuous averaging fuel per hour per segment of engine operation in L/s.

Returns
bool

5.5.1.23 getInstantFuelEconomy Method

bool getInstantFuelEconomy(float &InstantFuelEconomy);

Description

getInstantFuelEconomy(float &InstantFuelEconomy) will get current fuel economy at current vehicle velocity in Km/L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getInstantFuelEconomy(float InstantFuelEconomy);

Parameters

InstantFuelEconomy: the first parameter, float type, this is output parameter. It is the current fuel economy at current vehicle velocity in Km/L.

Returns
bool

5.5.1.24 getAvgFuelEconomy Method

bool getAvgFuelEconomy(float &AvgFuelEconomy);

Description

getAvgFuelEconomy(float &AvgFuelEconomy) will get average of instantaneous fuel economy for that segment of vehicle operation of interest in Km/L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAvgFuelEconomy(float AvgFuelEconomy);

Parameters

AvgFuelEconomy: the first parameter, float type, this is output parameter. It is the Average of instantaneous fuel economy for that segment of vehicle operation of interest in Km/L.

Returns
bool

5.5.1.25 getElectricalVoltage Method

bool getElectricalVoltage(float &BatteryVoltage);

Description

getElectricalVoltage(float &BatteryVoltage) will get electrical potential measured at the input of the electronic control unit supplied through a switching device. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getElectricalVoltage(float BatteryVoltage);

Parameters

BatteryVoltage: the first parameter, float type, this is output parameter. It is the electrical potential measured at the input of the electronic control unit supplied through a switching device.

Returns
bool

5.5.1.26 getRatedEnginePower Method

bool getRatedEnginePower(float &Power);

Description

getRatedEnginePower(float &Power) will get net brake power that the engine will deliver continuously, specified for a given application at a rated speed in KW. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getRatedEnginePower(float Power);

Parameters

Power: the first parameter, float type, this is output parameter. It is the net brake power that the engine will deliver continuously, specified for a given application at a rated speed in KW.

Returns
bool

5.5.1.27 getBatteryVoltage Method

bool getBatteryVoltage(float &BatteryVoltage);

Description

getBatteryVoltage(float &BatteryVoltage) will get measured electrical potential of the battery. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getBatteryVoltage(float BatteryVoltage);

Parameters

BatteryVoltage: the first parameter, float type, this is output parameter. It is the measured electrical potential of the battery.

Returns
bool

5.5.1.28 getAlternatorVoltage Method

bool getAlternatorVoltage(float &AlternatorVoltage);

Description

getAlternatorVoltage(float &AlternatorVoltage) will get measured electrical potential of the alternator. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAlternatorVoltage(float AlternatorVoltage);

Parameters

AlternatorVoltage: the first parameter, float type, this is output parameter. It is the measured electrical potential of the alternator.

Returns

bool

5.5.1.29 getAmbientTemp Method

bool getAmbientTemp(int &AmbientTemp);

Description

getAmbientTemp(int &AmbientTemp) will get temperature of air surrounding vehicle in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getAmbientTemp(int AmbientTemp);

Parameters

AmbientTemp: the first parameter,int type, this is output parameter. It is the temperature of air surrounding vehicle in Celsius degree.

Returns

bool

5.5.1.30 getCargoAmbientTemp Method

bool getCargoAmbientTemp(int &CargoTemp);

Description

getCargoAmbientTemp(int &CargoTemp) will get temperature of air inside vehicle container used to accommodate cargo in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708. getCargoAmbientTemp(int CargoTemp);

Parameters

CargoTemp: the first parameter,int type, this is output parameter. It is the temperature of air inside vehicle container used to accommodate cargo in Celsius degree.

Returns
bool

5.5.1.31 getRoadTemp Method

bool getRoadTemp(int &RoadTemp);

Description

getRoadTemp(int &RoadTemp) will get temperature of road surface over which vehicle is operating in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getRoadTemp(int RoadTemp);

Parameters

RoadTemp: the first parameter,int type, this is output parameter. It is the temperature of road surface over which vehicle is operating in Celsius degree.

Returns
bool

5.5.1.32 getCabInteriorTemp Method

bool getCabInteriorTemp(int &CabInteriorTemp);

Description

getCabInteriorTemp(int &CabInteriorTemp) will get temperature of air inside the part of the vehicle that encloses the driver and vehicle operating controls in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getCabInteriorTemp(int CabInteriorTemp);

Parameters

CabInteriorTemp: the first parameter,int type, this is output parameter. It is the temperature of air inside the part of the vehicle in Celsius degree.

Returns
bool

5.5.1.33 getInletTemp Method

bool getInletTemp(int &InletTemp);

Description

getInletTemp(int &InletTemp) will get temperature of air entering vehicle air induction system in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getInletTemp(int InletTemp);

Parameters

InletTemp: the first parameter,int type, this is output parameter. It is the temperature of air entering vehicle air induction system in Celsius degree.

Returns

bool

5.5.1.34 getFuelTemp Method

bool getFuelTemp(int &temp);

Description

getFuelTemp(int &temp) will get temperature of fuel entering injectors in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getFuelTemp(int temp);

Parameters

temp: the first parameter,int type, this is output parameter. It is the temperature of fuel entering injectors in Celsius degree.

Returns

bool

5.5.1.35 getOilTemp Method

bool getOilTemp(int &temp);

Description

getOilTemp(int &temp) will get temperature of engine lubricant in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getOilTemp(int temp);

Parameters

temp: the first parameter,int type, this is output parameter. It is the temperature of engine lubricant in Celsius degree.

Returns

bool

5.5.1.36 getCargoWeight Method

bool getCargoWeight(float &CargoW);

Description

getCargoWeight(float &CargoW) will get the force of gravity of freight carried in N. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getCargoWeight(float CargoW);

Parameters

CargoW: the first parameter,float type, this is output parameter. It is the force of gravity of freight carried in N.

Returns

bool

5.5.1.37 getEngineTripFuel Method

bool getEngineTripFuel(float &EngineTripFuel);

Description

getEngineTripFuel(float &EngineTripFuel) will get the fuel consumed during all or part of a journey in L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineTripFuel(float EngineTripFuel);

Parameters

EngineTripFuel: the first parameter,float type, this is output parameter. It is the fuel consumed during all or part of a journey in L.

Returns

bool

5.5.1.38 getEngineTotalFuelUsed Method

bool getEngineTotalFuelUsed(float &EngineTotalFuelUsed);

Description

getEngineTotalFuelUsed(float &EngineTotalFuelUsed) will get the accumulated amount of fuel used during vehicle operation in L. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineTotalFuelUsed(float EngineTotalFuelUsed);

Parameters

EngineTotalFuelUsed: the first parameter,float type, this is output parameter. It is the accumulated amount of fuel used during vehicle operation in L.

Returns
bool

5.5.1.39 getFuelRate Method

bool getFuelRate(float &FuelRate);

Description

getFuelRate(float &FuelRate) will get the amount of fuel consumed by engine per unit of time in L/s. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getFuelRate(float FuelRate);

Parameters

FuelRate: the first parameter,float type, this is output parameter. It is the amount of fuel consumed by engine per unit of time in L/s.

Returns
bool

5.5.1.40 getRatedEngineSpeed Method

bool getRatedEngineSpeed(float &RatedEngineSpeed);

Description

getRatedEngineSpeed(float &RatedEngineSpeed) will get the maximum governed rotational velocity of the engine crankshaft under full load conditions in rpm. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getRatedEngineSpeed(float RatedEngineSpeed);

Parameters

RatedEngineSpeed: the first parameter,float type, this is output parameter. It is the maximum governed rotational velocity of the engine crankshaft under full load conditions in rpm.

Returns
bool

5.5.1.41 getEngineSpeed Method

bool getEngineSpeed(int &EngineSpeed);

Description

getEngineSpeed(int &EngineSpeed) will get the rotational velocity of crankshaft in rpm. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getEngineSpeed(int EngineSpeed);

Parameters

EngineSpeed: the first parameter,int type, this is output parameter. It is the rotational velocity of crankshaft in rpm.

Returns
bool

5.5.1.42 getIntakeManifoldTemp Method

bool getIntakeManifoldTemp(int &IntakeManifoldTemp);

Description

getIntakeManifoldTemp(int &IntakeManifoldTemp) will get the temperature of precombustion air found in intake manifold of engine air supply system in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getIntakeManifoldTemp(int IntakeManifoldTemp);

Parameters

IntakeManifoldTemp: the first parameter,int type, this is output parameter. It is the temperature of precombustion air found in intake manifold of engine air supply system in Celsius degree.

Returns

bool

5.5.1.43 getPowerTakeoffStatus Method

```
bool getPowerTakeoffStatus(bool &PTOModeActive,  bool &ClutchSwitchOn,
                           bool &BrakeSwitchOn,  bool &AccelSwitchOn,
                           bool &ResumeSwitchOn, bool &CoastSwitchOn,
                           bool &SetSwitchOn, bool &PTOControlSwitchOn);
```

Description

getPowerTakeoffStatus will get the state of the system used to transmit engine power to auxiliary equipment. It will return true if success, otherwise return false.

Syntax

```
DFL168A.J1708.getPowerTakeoffStatus(bool PTOModeActive,  bool ClutchSwitchOn, bool
                                    BrakeSwitchOn,
                                    bool AccelSwitchOn, bool ResumeSwitchOn, bool
                                    CoastSwitchOn,
                                    bool SetSwitchOn, bool PTOControlSwitchOn);
```

Parameters

PTOModeActive: the first parameter,bool type, this is output parameter. It is PTO mode. True means "active", False means "not active"

ClutchSwitchOn: the second parameter,bool type, this is output parameter. It is clutch switch. True means "On", False means "Off"

BrakeSwitchOn: the third parameter,bool type, this is output parameter. It is brake switch. True means "On", False means "Off"

AccelSwitchOn: the fourth parameter,bool type, this is output parameter. It is accel switch. True means "On", False means "Off"

ResumeSwitchOn: the 5th parameter,bool type, this is output parameter. It is resume switch. True means "On", False means "Off"

CoastSwitchOn: the 6th parameter,bool type, this is output parameter. It is coast switch. True means "On", False means "Off"

SetSwitchOn: the 7th parameter,bool type, this is output parameter. It is set switch. True means "On", False means "Off"

PTOControlSwitchOn: the 8th parameter,bool type, this is output parameter. It is PTO control switch. True means "On", False means "Off"

Returns
bool

5.5.1.44 getTripDistance Method

bool getTripDistance(float &TripDistance);

Description

getTripDistance(float &TripDistance) will get the distance traveled during all or part of a journey in Km. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTripDistance(float TripDistance);

Parameters

TripDistance: the first parameter,float type, this is output parameter. It is the distance traveled during all or part of a journey in Km.

Returns
bool

5.5.1.45 getTotalDistance Method

bool getTotalDistance(float &TotalDistance);

Description

getTotalDistance(float &TotalDistance) will get the accumulated distance travelled by vehicle during its operation in Km. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTotalDistance(float TotalDistance);

Parameters

TotalDistance: the first parameter,float type, this is output parameter. It is the accumulated distance travelled by vehicle during its operation in Km.

Returns
bool

5.5.1.46 getTotalEngineHours Method

bool getTotalEngineHours(float &TotalEngineHours);

Description

getTotalEngineHours(float &TotalEngineHours) will get the accumulated time of operation of engine in hours. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTotalEngineHours(float TotalEngineHours);

Parameters

TotalEngineHours: the first parameter,float type, this is output parameter. It is the accumulated time of operation of engine in hours.

Returns

bool

5.5.1.47 getTotalEngineRevolutions Method

bool getTotalEngineRevolutions(float &TotalEngineRevolutions);

Description

getTotalEngineRevolutions(float &TotalEngineRevolutions) will get the accumulated number of revolutions of engine crankshaft during its operation. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getTotalEngineRevolutions(float TotalEngineRevolutions);

Parameters

TotalEngineRevolutions: the first parameter,float type, this is output parameter. It is the accumulated number of revolutions of engine crankshaft during its operation.

Returns

bool

5.5.1.48 getVIN Method

bool getVIN(String &VIN);

Description

getVIN(String &VIN) will get the Vehicle Identification Number (VIN) as assigned by the vehicle manufacturer. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getVIN(String VIN);

Parameters

VIN: the first parameter,String type, this is output parameter. It is the Vehicle Identification Number.

Returns
bool

5.5.1.49 getDTC Method

```
bool getDTC(byte &DTC_Num, byte &MID,       int (&PID_SID)[8],
                bool (&IsPID)[8],  byte (&FMI)[8],   bool (& IsActive)[8],
                bool (&OccurrenceExist)[8]          ,byte (&OccurrenceCount)[8]);
```

Description

getDTC will get the diagnostic code and occurrence count. It will return true if success, otherwise return false.

Syntax

```
DFL168A.J1708.getDTC(byte DTC_Num,       byte MID,       int PID_SID[8],
                                bool IsPID[8],          byte FMI[8],     bool IsActive)[8],
                                bool OccurrenceExist[8],       byte OccurrenceCount[8]);
```

Parameters

DTC_Num: the first parameter, byte type, this is output parameter. It is the quantity of DTC. The first DTC will be in index 0 of array, The second will be in index 1 of array, ..., Maximum of 8 DTCs

MID: the second parameter, byte type, this is output parameter. It is MID of DTC.

PID_SID[8]: the third parameter, int array type, this is output parameter. It is SID or PID of a standard diagnostic code. The next parameter will decide it is SID or PID.

IsPID[8]: the fourth parameter, bool array type, this is output parameter. It will tell us whether above parameter PID_SID[8] is PID. True means "PID", False means "SID"

FMI[8]: the 5th parameter, byte array type, this is output parameter. It will tell us Failure mode identifier (FMI) of a standard diagnostic code.

IsActive)[8]: the 6th parameter, bool array type, this is output parameter. It will tell us whether fault is active. True means "active", False means "inactive"

OccurrenceExist[8]: the 7th parameter, bool array type, this is output parameter. It will tell whether the next parameter OccurrenceCount exists. True means "exist", False means "not exist"

OccurrenceCount[8]: the 8th parameter, byte array type, this is output parameter. It will tell occurrence count for this diagnostic code

Returns
bool

5.5.1.50 clearDTC Method

bool clearDTC(byte MID,int PID_SID, bool IsPID);

Description

clearDTC(byte MID,int PID_SID, bool IsPID) will be used to clear diagnostic codes on the device with
the given MID, PID or SID. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.clearDTC(byte MID,int PID_SID, bool IsPID);

Parameters

MID: the first parameter,byte type, this is input parameter. It is the MID of device which some special DTC will be cleared..

PID_SID: the second parameter,int type, this is input parameter. It is the PID or SID which will be cleared. The next parameter will decide PID or SID.

IsPID: the third parameter,bool type, this is input parameter. It is used to identify the above parameter PID_SID. True means "PID", False means "SID"

Returns
bool

5.5.1.51 getFaultDescription Method

bool getFaultDescription(byte MID,int PID_SID, bool IsPID,byte FMI, String & FaultDescription);

Description

getFaultDescription will be used to get DTC description string on the device with
the given MID, PID or SID, and FMI. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getFaultDescription(byte MID,int PID_SID, bool IsPID,byte FMI, String FaultDescription);

Parameters

MID: the first parameter,byte type, this is input parameter. It is the MID of device which some special PID or SID and FMI will be described..

PID_SID: the second parameter,int type, this is input parameter. It is the PID or SID. The next parameter will decide PID or SID.

IsPID: the third parameter,bool type, this is input parameter. It is used to identify the above parameter PID_SID. True means "PID", False means "SID"

FMI: the 4th parameter,byte type, this is input parameter. It is FMI of DTC which needs to be described.

FaultDescription: the 5th parameter,String type, this is output parameter. It is the string of described .DTC.

Returns

bool

5.5.1.52 getPIDSIDDescription Method

bool getPIDSIDDescription(byte MID,int PID_SID, bool IsPID, String & PID_SID_Description);

Description

getPIDSIDDescription will be used to get PID or SID description string on the device with the given MID. It will return true if success, otherwise return false.

Syntax

DFL168A.J1708.getPIDSIDDescription(byte MID,int PID_SID, bool IsPID, String & PID_SID_Description);

Parameters

MID: the first parameter,byte type, this is input parameter. It is the MID of device which some special PID or SID will be described..

PID_SID: the second parameter,int type, this is input parameter. It is the PID or SID. The next parameter will decide PID or SID.

IsPID: the third parameter,bool type, this is input parameter. It is used to identify the above parameter PID_SID. True means "PID", False means "SID"

PID_SID_Description: the 4th parameter,String type, this is output parameter. It is the string of described PID or SID.

Returns

bool

5.6 Inner Class ISO15765

5.6.1 Methods

bool getCoolantTemperature(int &temp);

bool getEngineSpeed(int &EngineSpeed);

```
bool getVehicleSpeed(float &VehicleSpeed);
bool getIntakeManifoldPressure(int &IntakeManifoldPressure);
bool getFuelSystemStatus(bool &A_Openloop,     bool &A_Closedloop,
                         bool &A_OpenloopByDriving_Con,  bool &A_OpenloopByFault,
                         bool &A_ClosedloopButFault,  bool &B_Openloop,
                         bool &B_Closedloop,   bool &B_OpenloopByDriving_Con,
                         bool &B_OpenloopByFault,bool &B_ClosedloopButFault);
bool getCalculatedLoadValue(int &CalculatedLoad);
bool getShortTermFuelTrimBank13(float &Bank1,float &Bank3);
bool getLongTermFuelTrimBank13(float &Bank1,float &Bank3);
bool getShortTermFuelTrimBank24(float &Bank2,float &Bank4);
bool getLongTermFuelTrimBank24(float &Bank2,float &Bank4);
bool getIgnitionTimingAdvance(float &Angle);
bool getIntakeAirTemperature(int &temp);
bool getAirFlowRateFrmMAF(float &FlowRate);
bool getAbsThrottlePosition(float &Percent);
bool getOxygenSensorLocation(bool &Bank1_Sensor1Present, bool &Bank1_Sensor2Present,
                             bool &Bank1_Sensor3Present,  bool &Bank1_Sensor4Present,
                             bool &Bank3_Sensor1Present,  bool &Bank3_Sensor2Present,
                             bool &Bank2_Sensor1Present,  bool &Bank2_Sensor2Present,
                             bool &Bank2_Sensor3Present,  bool &Bank2_Sensor4Present,
                            bool &Bank4_Sensor1Present,   bool &Bank4_Sensor2Present);
bool getBank1OSensor1Voltage(float &OutVoltage);
bool getBank1OSensor2Voltage(float &OutVoltage);
bool getBank1OSensor3Voltage(float &OutVoltage);
bool getBank1OSensor4Voltage(float &OutVoltage);
bool getBank2OSensor1Voltage(float &OutVoltage);
bool getBank2OSensor2Voltage(float &OutVoltage);
bool getBank2OSensor3Voltage(float &OutVoltage);
bool getBank2OSensor4Voltage(float &OutVoltage);
bool getBank3OSensor1Voltage(float &OutVoltage);
bool getBank3OSensor2Voltage(float &OutVoltage);
bool getBank4OSensor1Voltage(float &OutVoltage);
bool getBank4OSensor2Voltage(float &OutVoltage);
bool getOBDCertified(String &OBD);
bool getTimeSinceEngineStart(unsigned int &TotalTime);
bool getDistanceTraveledMIL(unsigned int &Distance);
bool getFuelRailPressure(float &Pressure);
bool getFuelLevelInput(float &Percent);
```

```
bool getDistanceTraveledSinceDTC_Clear(unsigned int &Distance);
bool getBarometricPressure(int &Pressure);
bool getControlModuleVoltage(float &Voltage);
bool getRelativeThrottlePosition(float &Percent);
bool getAmbientTemp(int &AmbientTemp);
bool getCommandedThrottleActuatorControl(float &Percent);
bool getEngineRunTimeMIL(unsigned int &TotalTime);
bool getEngineRunTimeSinceDTC_Clear(unsigned int &TotalTime);
bool getTypeOfFuelUsedCurrently(String & FuelType);
bool getRelativeAcceleratorPedalPosition(float &Percent);
bool getHybridBatteryPackRemainingLife(float &Percent);
bool getEngineOilTemperature(int &Tem);
bool getFuelRate(float &FuelRate);
bool getActualEngineTorque(int &ActualEngineTorque);
bool getMILStatus(bool &MIL_IS_ON);
bool getEngineRunTime(unsigned long &TotalEngineRunTime,
                      unsigned long &TotalIdleRunTime,
                      unsigned long &TotalRunTimeWithPTOActive);
bool getVIN(String &VIN);
bool getDTC(byte &DTC_Num,String ( &DTC)[8]);
boolean clearDTC();
```

5.6.1.1 getCoolantTemperature Method

```
bool getCoolantTemperature(int &temp);
```

Description

getCoolantTemperature(int &temp) will get engine coolant temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

```
DFL168A.ISO15765.getCoolantTemperature(int temp);
```

Parameters

temp: the first parameter, int type, this is output parameter. It is engine coolant temperature in Celsius degree.

Returns

bool

5.6.1.2 getEngineSpeed Method

bool getEngineSpeed(int &EngineSpeed);

Description

getEngineSpeed(int &EngineSpeed) will get revolutions per minute of the engine crankshaft. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getEngineSpeed(int EngineSpeed);

Parameters

EngineSpeed: the first parameter, int type, this is output parameter. It is the revolutions per minute of the engine crankshaft.

Returns
bool

5.6.1.3 getVehicleSpeed Method

bool getVehicleSpeed(float &VehicleSpeed);

Description

getVehicleSpeed(float &VehicleSpeed) will get vehicle road speed in Km/h. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getVehicleSpeed(float VehicleSpeed);

Parameters

VehicleSpeed: the first parameter, float type, this is output parameter. It is the vehicle road speed in Km/h.

Returns
bool

5.6.1.4 getIntakeManifoldPressure Method

bool getIntakeManifoldPressure(int &IntakeManifoldPressure);

Description

getIntakeManifoldPressure(int &IntakeManifoldPressure) will get manifold pressure derived from a Manifold Absolute Pressure sensor in kPa. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getIntakeManifoldPressure(int IntakeManifoldPressure);

Parameters

IntakeManifoldPressure: the first parameter, int type, this is output parameter. It is the manifold pressure in kPa.

Returns

bool

5.6.1.5 getFuelSystemStatus Method

```
bool getFuelSystemStatus(bool &A_Openloop,                    bool &A_Closedloop,
                         bool &A_OpenloopByDriving_Con, bool &A_OpenloopByFault,
                         bool &A_ClosedloopButFault,    bool &B_Openloop,
                         bool &B_Closedloop,            bool &B_OpenloopByDriving_Con,
                         bool &B_OpenloopByFault     ,bool &B_ClosedloopButFault);
```

Description

getFuelSystemStatus will get fuel system status. It will return true if success, otherwise return false.

Syntax

```
DFL168A.ISO15765.getFuelSystemStatus(bool A_Openloop,      bool A_Closedloop,
                                     bool A_OpenloopByDriving_Con, bool
                                     A_OpenloopByFault,
                                     bool A_ClosedloopButFault,    bool B_Openloop,
                                     bool B_Closedloop,            bool
                                     B_OpenloopByDriving_Con,
                                     bool B_OpenloopByFault     ,bool
                                     B_ClosedloopButFault);
```

Parameters

A_Openloop: the first parameter, bool type, this is output parameter. True means "fuel system 1: Open loop" , It means that it has not yet satisfied conditions to go closed loop

A_Closedloop: the second parameter, bool type, this is output parameter. True means " fuel system 1: Closed loop" , It means that it is using oxygen sensor(s) as feedback for fuel control.

A_OpenloopByDriving_Con: the third parameter, bool type, this is output parameter. True means "fuel system1: Open loop due to driving conditions (e.g.power enrichment, deceleration enleanment)"

A_OpenloopByFault: the 4th parameter, bool type, this is output parameter. True means "fuel system1: Open loop - due to detected system fault"

A_ClosedloopButFault: the 5th parameter, bool type, this is output parameter. True means "fuel system1: Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control"

B_Openloop: the 6th parameter, bool type, this is output parameter. True means "fuel system 2: Open loop" , It means that it has not yet satisfied conditions to go closed loop

B_Closedloop: the 7th parameter, bool type, this is output parameter. True means " fuel system 2: Closed loop" , It means that it is using oxygen sensor(s) as feedback for fuel control.

B_OpenloopByDriving_Con: the 8th parameter, bool type, this is output parameter. True means "fuel system2: Open loop due to driving conditions (e.g.power enrichment, deceleration enleanment)"

B_OpenloopByFault: the 9th parameter, bool type, this is output parameter. True means "fuel system2: Open loop - due to detected system fault"

B_ClosedloopButFault: the 10th parameter, bool type, this is output parameter. True means "fuel system2: Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control"

Returns

bool

5.6.1.6 getCalculatedLoadValue Method

bool getCalculatedLoadValue(int &CalculatedLoad);

Description

getCalculatedLoadValue(int &CalculatedLoad) will get calculated LOAD Value in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getCalculatedLoadValue(int CalculatedLoad);

Parameters

CalculatedLoad: the first parameter, int type, this is output parameter. It is the calculated LOAD Value in percentage.

Returns
bool

5.6.1.7 getShortTermFuelTrimBank13 Method

bool getShortTermFuelTrimBank13(float &Bank1,float &Bank3);

Description

getShortTermFuelTrimBank13(float &Bank1,float &Bank3) will get short term fuel trim in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getShortTermFuelTrimBank13(float Bank1,float Bank3);

Parameters

Bank1: the first parameter, float type, this is output parameter. It is the short term fuel trim - Bank 1 in percentage.

Bank3: the first parameter, float type, this is output parameter. It is the short term fuel trim - Bank 3 in percentage.

Returns
bool

5.6.1.8 getLongTermFuelTrimBank13 Method

bool getLongTermFuelTrimBank13(float &Bank1,float &Bank3);

Description

getLongTermFuelTrimBank13(float &Bank1,float &Bank3) will get long term fuel trim in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getLongTermFuelTrimBank13(float Bank1,float Bank3);

Parameters

Bank1: the first parameter, float type, this is output parameter. It is the long term fuel trim - Bank 1 in percentage.

Bank3: the first parameter, float type, this is output parameter. It is the long term fuel trim - Bank 3 in percentage.

Returns
bool

5.6.1.9 getShortTermFuelTrimBank24 Method

bool getShortTermFuelTrimBank24(float &Bank2,float &Bank4);

Description

getShortTermFuelTrimBank24(float &Bank2,float &Bank4) will get short term fuel trim in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getShortTermFuelTrimBank24(float Bank2,float Bank4);

Parameters

Bank2: the first parameter, float type, this is output parameter. It is the short term fuel trim - Bank 2 in percentage.

Bank4: the first parameter, float type, this is output parameter. It is the short term fuel trim - Bank 4 in percentage.

Returns
bool

5.6.1.10 getLongTermFuelTrimBank24 Method

bool getLongTermFuelTrimBank24(float &Bank2,float &Bank4);

Description

getLongTermFuelTrimBank24(float &Bank2,float &Bank4) will get long term fuel trim in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getLongTermFuelTrimBank24(float Bank2,float Bank4);

Parameters

Bank2: the first parameter, float type, this is output parameter. It is the long term fuel trim - Bank 2 in percentage.

Bank4: the first parameter, float type, this is output parameter. It is the long term fuel trim - Bank 4 in percentage.

Returns
bool

5.6.1.11 getIgnitionTimingAdvance Method

bool getIgnitionTimingAdvance(float &Angle);

Description

getIgnitionTimingAdvance(float &Angle) will get Ignition timing spark advance in degrees before top dead center (°BTDC) for #1 cylinder (not including mechanical advance). It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getIgnitionTimingAdvance(float Angle);

Parameters

Angle: the first parameter, float type, this is output parameter. It is the Ignition Timing Advance for #1 Cylinder in degree.

Returns

bool

5.6.1.12 getIntakeAirTemperature Method

bool getIntakeAirTemperature(int &temp);

Description

getIntakeAirTemperature(int &temp) will get intake manifold air temperature in Celsius degree. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getIntakeAirTemperature(int temp);

Parameters

temp: the first parameter, int type, this is output parameter. It is the intake manifold air temperature in Celsius degree.

Returns

bool

5.6.1.13 getAirFlowRateFrmMAF Method

bool getAirFlowRateFrmMAF(float &FlowRate);

Description

getAirFlowRateFrmMAF(float &FlowRate) will get air flow rate from mass air flow sensor in g/s. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getAirFlowRateFrmMAF(float FlowRate);

Parameters

FlowRate: the first parameter, float type, this is output parameter. It is the air flow rate from mass air flow sensor in g/s.

Returns

bool

5.6.1.14 getAbsThrottlePosition Method

bool getAbsThrottlePosition(float &Percent);

Description

getAbsThrottlePosition(float &Percent) will get absolute throttle position in percentage. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getAbsThrottlePosition(float Percent);

Parameters

Percent: the first parameter, float type, this is output parameter. It is the absolute throttle position in percentage.

Returns

bool

5.6.1.15 getOxygenSensorLocation Method

bool getOxygenSensorLocation(bool &Bank1_Sensor1Present, bool &Bank1_Sensor2Present,
bool &Bank1_Sensor3Present, bool &Bank1_Sensor4Present,
bool &Bank3_Sensor1Present, bool &Bank3_Sensor2Present,
bool &Bank2_Sensor1Present, bool &Bank2_Sensor2Present,
bool &Bank2_Sensor3Present, bool &Bank2_Sensor4Present,
bool &Bank4_Sensor1Present, bool &Bank4_Sensor2Present);

Description

getOxygenSensorLocation will get location of oxygen sensors. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.bool getOxygenSensorLocation(bool Bank1_Sensor1Present, bool Bank1_Sensor2Present,

```
                                    bool Bank1_Sensor3Present,   bool
                                    Bank1_Sensor4Present,
                                    bool Bank3_Sensor1Present,   bool
                                    Bank3_Sensor2Present,
                                    bool Bank2_Sensor1Present,   bool
                                    Bank2_Sensor2Present,
                                    bool Bank2_Sensor3Present,   bool
                                    Bank2_Sensor4Present,
                                    bool Bank4_Sensor1Present,   bool
                                    Bank4_Sensor2Present);
```

Parameters

Bank1_Sensor1Present: the first parameter, bool type, this is output parameter. True mans Bank 1 - Sensor 1 present at that location.

Bank1_Sensor2Present: the 2nd parameter, bool type, this is output parameter. True mans Bank 1 - Sensor 2 present at that location.

Bank1_Sensor3Present: the 3rd parameter, bool type, this is output parameter. True mans Bank 1 - Sensor 3 present at that location.

Bank1_Sensor4Present: the 4th parameter, bool type, this is output parameter. True mans Bank 1 - Sensor 4 present at that location.

Bank3_Sensor1Present: the 5th parameter, bool type, this is output parameter. True mans Bank 3 - Sensor 1 present at that location.

Bank3_Sensor2Present: the 6th parameter, bool type, this is output parameter. True mans Bank 3 - Sensor 2 present at that location.

Bank2_Sensor1Present: the 7th parameter, bool type, this is output parameter. True mans Bank 2 - Sensor 1 present at that location.

Bank2_Sensor2Present: the 8th parameter, bool type, this is output parameter. True mans Bank 2 - Sensor 2 present at that location.

Bank2_Sensor3Present: the 9th parameter, bool type, this is output parameter. True mans Bank 2 - Sensor 3 present at that location.

Bank2_Sensor4Present: the 10th parameter, bool type, this is output parameter. True mans Bank 2 - Sensor 4 present at that location.

Bank4_Sensor1Present: the 11th parameter, bool type, this is output parameter. True mans Bank 4 - Sensor 1 present at that location.

Bank4_Sensor2Present: the 12th parameter, bool type, this is output parameter. True mans Bank 4 - Sensor 2 present at that location.

Returns

bool

5.6.1.16 getBank1OSensor1Voltage Method

bool getBank1OSensor1Voltage(float &OutVoltage);

Description

getBank1OSensor1Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 1 – Sensor 1. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank1OSensor1Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 1 – Sensor 1

Returns

bool.

5.6.1.17 getBank1OSensor2Voltage Method

bool getBank1OSensor2Voltage(float &OutVoltage);

Description

getBank1OSensor2Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 1 – Sensor 2. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank1OSensor2Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen

sensor for Bank 1 – Sensor 2

Returns
bool.

5.6.1.18 getBank1OSensor3Voltage Method

bool getBank1OSensor3Voltage(float &OutVoltage);

Description

getBank1OSensor3Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 1 – Sensor 3. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank1OSensor3Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 1 – Sensor 3

Returns
bool.

5.6.1.19 getBank1OSensor4Voltage Method

bool getBank1OSensor4Voltage(float &OutVoltage);

Description

getBank1OSensor4Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 1 – Sensor 4. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank1OSensor4Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 1 – Sensor 4

Returns
bool.

5.6.1.20 getBank2OSensor1Voltage Method

bool getBank2OSensor1Voltage(float &OutVoltage);

Description

getBank2OSensor1Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 2 – Sensor

1. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank2OSensor1Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 2 – Sensor 1

Returns

bool.

5.6.1.21 getBank2OSensor2Voltage Method

bool getBank2OSensor2Voltage(float &OutVoltage);

Description

getBank2OSensor2Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 2 – Sensor 2. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank2OSensor2Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 2 – Sensor 2

Returns

bool.

5.6.1.22 getBank2OSensor3Voltage Method

bool getBank2OSensor3Voltage(float &OutVoltage);

Description

getBank2OSensor3Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 2 – Sensor 3. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank2OSensor3Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 2 – Sensor 3

Returns
bool.

5.6.1.23 getBank2OSensor4Voltage Method

bool getBank2OSensor4Voltage(float &OutVoltage);

Description

getBank2OSensor4Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 2 – Sensor 4. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank2OSensor4Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 2 – Sensor 4

Returns
bool.

5.6.1.24 getBank3OSensor1Voltage Method

bool getBank3OSensor1Voltage(float &OutVoltage);

Description

getBank3OSensor1Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 3 – Sensor 1. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank3OSensor1Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 3 – Sensor 1

Returns
bool.

5.6.1.25 getBank3OSensor2Voltage Method

bool getBank3OSensor2Voltage(float &OutVoltage);

Description

getBank3OSensor2Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 3 – Sensor 2. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank3OSensor2Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 3 – Sensor 2

Returns
bool.

5.6.1.26 getBank4OSensor1Voltage Method

bool getBank4OSensor1Voltage(float &OutVoltage);

Description

getBank4OSensor1Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 4 – Sensor 1. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank4OSensor1Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 4 – Sensor 1

Returns
bool.

5.6.1.27 getBank4OSensor2Voltage Method

bool getBank4OSensor2Voltage(float &OutVoltage);

Description

getBank4OSensor2Voltage(float &OutVoltage) will get 0 to 1 volt oxygen sensor for Bank 4 – Sensor 2. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBank4OSensor2Voltage(float OutVoltage);

Parameters

OutVoltage: the first parameter, float type, this is output parameter. It is the voltage value of oxygen sensor for Bank 4 – Sensor 2

Returns
bool.

5.6.1.28 getOBDCertified Method

bool getOBDCertified(String &OBD);

Description

getOBDCertified(String &OBD) will get OBD requirements String to which vehicle or engine is certified.. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getOBDCertified(String OBD);

Parameters

OBD: the first parameter, String type, this is output parameter. It is the OBD certified String

Returns

bool.

5.6.1.29 getTimeSinceEngineStart Method

bool getTimeSinceEngineStart(unsigned int &TotalTime);

Description

getTimeSinceEngineStart(unsigned int &TotalTime) will get time Since Engine Start in seconds. For non-hybrid vehicles, TotalTime shall increment while the engine is running. It shall freeze if the engine stalls. TotalTime shall be reset to zero during every control module power-up and when entering the key-on, engine off position. TotalTime is limited to 65535 seconds and shall not wrap around to zero. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getTimeSinceEngineStart(unsigned int TotalTime);

Parameters

TotalTime: the first parameter, unsigned int type, this is output parameter. It is the time (Seconds) since engine start

Returns

bool.

5.6.1.30 getDistanceTraveledMIL Method

bool getDistanceTraveledMIL(unsigned int &Distance);

Description

getDistanceTraveledMIL(unsigned int &Distance) will get the distance (Km) traveled while MIL is activated. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getDistanceTraveledMIL(unsigned int Distance);

Parameters

Distance: the first parameter, unsigned int type, this is output parameter. It is the distance (Km) traveled while MIL is activated

Returns
bool.

5.6.1.31 getFuelRailPressure Method

bool getFuelRailPressure(float &Pressure);

Description

getFuelRailPressure(float &Pressure) will get the fuel rail pressure (kPa) at the engine when the reading is referenced to atmosphere (gage pressure). This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getFuelRailPressure(float Pressure);

Parameters

Pressure: the first parameter, float type, this is output parameter. It is the fuel rail pressure in kPa.

Returns
bool.

5.6.1.32 getFuelLevelInput Method

bool getFuelLevelInput(float &Percent);

Description

getFuelLevelInput(float &Percent) will get the nominal fuel tank liquid fill capacity as a percent of maximum. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getFuelLevelInput(float Percent);

Parameters

Percent: the first parameter, float type, this is output parameter. It is the fuel level input in percentage.

Returns
bool.

5.6.1.33 getDistanceTraveledSinceDTC_Clear Method

bool getDistanceTraveledSinceDTC_Clear(unsigned int &Distance);

Description

getDistanceTraveledSinceDTC_Clear(unsigned int &Distance) will get the distance (Km) accumulated since DTCs were cleared (via external test equipment or possibly, a battery disconnect). This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getDistanceTraveledSinceDTC_Clear(unsigned int Distance);

Parameters

Distance: the first parameter, unsigned int type, this is output parameter. It is the distance (Km) traveled since
DTCs cleared

Returns
bool.

5.6.1.34 getBarometricPressure Method

bool getBarometricPressure(int &Pressure);

Description

getBarometricPressure(int &Pressure) will get the barometric pressure in kPa. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getBarometricPressure(int Pressure);

Parameters

Pressure: the first parameter, float type, this is output parameter. It is the barometric pressure in kPa.

Returns
bool.

5.6.1.35 getControlModuleVoltage Method

bool getControlModuleVoltage(float &Voltage);

Description

getControlModuleVoltage(float &Voltage) will get the control module voltage. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getControlModuleVoltage(float Voltage);

Parameters

Voltage: the first parameter, float type, this is output parameter. It is the control module voltage.

Returns
bool.

5.6.1.36 getRelativeThrottlePosition Method

bool getRelativeThrottlePosition(float &Percent);

Description

getRelativeThrottlePosition(float &Percent) will get the relative throttle position in percentage. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getRelativeThrottlePosition(float Percent);

Parameters

Percent: the first parameter, float type, this is output parameter. It is the relative throttle position in percentage.

Returns
bool.

5.6.1.37 getAmbientTemp Method

bool getAmbientTemp(int &AmbientTemp);

Description

getAmbientTemp(int &AmbientTemp) will get the ambient air temperature in Celsius degree. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getAmbientTemp(int AmbientTemp);

Parameters

AmbientTemp: the first parameter, int type, this is output parameter. It is the ambient air temperature in Celsius degree.

Returns
bool.

5.6.1.38 getCommandedThrottleActuatorControl Method

bool getCommandedThrottleActuatorControl(float &Percent);

Description

getCommandedThrottleActuatorControl(float &Percent) will get the commanded throttle actuator control in percentage. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getCommandedThrottleActuatorControl(float Percent);

Parameters

Percent: the first parameter, float type, this is output parameter. It is the commanded throttle actuator control in percentage.

Returns

bool.

5.6.1.39 getEngineRunTimeMIL Method

bool getEngineRunTimeMIL(unsigned int &TotalTime);

Description

getEngineRunTimeMIL(unsigned int &TotalTime) will get the engine run time (minutes) while MIL is activated. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getEngineRunTimeMIL(unsigned int TotalTime);

Parameters

TotalTime: the first parameter, unsigned int type, this is output parameter. It is the engine run time (minutes) while MIL is activated.

Returns

bool.

5.6.1.40 getEngineRunTimeSinceDTC_Clear Method

bool getEngineRunTimeSinceDTC_Clear(unsigned int &TotalTime);

Description

getEngineRunTimeSinceDTC_Clear(unsigned int &TotalTime) will get the engine run time (minutes) since DTCs cleared. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getEngineRunTimeSinceDTC_Clear(unsigned int TotalTime);

Parameters

TotalTime: the first parameter, unsigned int type, this is output parameter. It is the engine run time (minutes) since DTCs cleared.

Returns

bool.

5.6.1.41 getTypeOfFuelUsedCurrently Method

bool getTypeOfFuelUsedCurrently(String & FuelType);

Description

getTypeOfFuelUsedCurrently(String & FuelType) will get the type of fuel currently being utilized by the vehicle. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getTypeOfFuelUsedCurrently(String FuelType);

Parameters

FuelType: the first parameter, String type, this is output parameter. It is the type of fuel currently being utilized by the vehicle.

Returns

bool.

5.6.1.42 getRelativeAcceleratorPedalPosition Method

bool getRelativeAcceleratorPedalPosition(float &Percent);

Description

getRelativeAcceleratorPedalPosition(float &Percent) will get the relative accelerator pedal position in percentage. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getRelativeAcceleratorPedalPosition(float Percent);

Parameters

Percent: the first parameter, float type, this is output parameter. It is the relative accelerator pedal position in percentage.

Returns

bool.

5.6.1.43 getHybridBatteryPackRemainingLife Method

bool getHybridBatteryPackRemainingLife(float &Percent);

Description

getHybridBatteryPackRemainingLife(float &Percent) will get the percent remaining life for the hybrid battery pack. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getHybridBatteryPackRemainingLife(float Percent);

Parameters

Percent: the first parameter, float type, this is output parameter. It is the percent remaining life for the hybrid battery pack.

Returns

bool.

5.6.1.44 getEngineOilTemperature Method

bool getEngineOilTemperature(int &Tem);

Description

getEngineOilTemperature(int &Tem) will get the engine oil temperature in Celsius degree. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getEngineOilTemperature(int Tem);

Parameters

Tem: the first parameter, int type, this is output parameter. It is the engine oil temperature in Celsius degree.

Returns

bool.

5.6.1.45 getFuelRate Method

bool getFuelRate(float &FuelRate);

Description

getFuelRate(float &FuelRate) will get the amount of fuel consumed by engine per unit of time in liters per hour. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getFuelRate(float FuelRate);

Parameters

FuelRate: the first parameter, float type, this is output parameter. It is the amount of fuel consumed by engine per unit of time in liters per hour.

Returns

bool.

5.6.1.46 getActualEngineTorque Method

bool getActualEngineTorque(int &ActualEngineTorque);

Description

getActualEngineTorque(int &ActualEngineTorque) will get the actual engine - percent torque. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getActualEngineTorque(int ActualEngineTorque);

Parameters

ActualEngineTorque: the first parameter, int type, this is output parameter. It is the actual engine - percent torque.

Returns

bool.

5.6.1.47 getMILStatus Method

bool getMILStatus(bool &MIL_IS_ON);

Description

getMILStatus(bool &MIL_IS_ON) will get the Malfunction Indicator Lamp (MIL) status. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getMILStatus(bool MIL_IS_ON);

Parameters

MIL_IS_ON: the first parameter, bool type, this is output parameter. True means MIL is ON

Returns

bool.

5.6.1.48 getEngineRunTime Method

bool getEngineRunTime(unsigned long &TotalEngineRunTime, unsigned long &TotalIdleRunTime, unsigned long &TotalRunTimeWithPTOActive);

Description

getEngineRunTime will get the engine run time. This function will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getEngineRunTime(unsigned long TotalEngineRunTime, unsigned long TotalIdleRunTime,
unsigned long TotalRunTimeWithPTOActive);

Parameters

TotalEngineRunTime: the first parameter, unsigned long type, this is output parameter. It is the total engine run time (Seconds) for the life of vehicle. It shall increment while the engine is running. It shall freeze if the engine stalls.It shall never be reset to zero.

TotalIdleRunTime: the first parameter, unsigned long type, this is output parameter. It is the the total engine idle time (Seconds) for the life of vehicle. It shall increment while the engine is running at closed throttle/closed pedal and vehicle speed is less than 5 kph. It shall freeze if the engine stalls or the engine is no longer at idle. It shall never be reset to zero..

TotalRunTimeWithPTOActive: the first parameter, unsigned long type, this is output parameter. It is the total engine run time (Seconds) with PTO engaged for the life of vehicle. It shall increment while the engine is running with PTO engaged. It shall freeze if the engine stalls. It shall never be reset to zero..

Returns

bool.

5.6.1.49 getVIN Method

bool getVIN(String &VIN);

Description

getVIN(String &VIN) will get the Vehicle Identification Number (VIN) as assigned by the vehicle manufacturer. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getVIN(String VIN);

Parameters

VIN: the first parameter,String type, this is output parameter. It is the Vehicle Identification Number.

Returns
bool.

5.6.1.50 getDTC Method

bool getDTC(byte &DTC_Num,String (&DTC)[8]);

Description

getDTC(byte &DTC_Num,String (&DTC)[8]) will get all DTCs. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getDTC(byte DTC_Num,String DTC[8]);

Parameters

DTC_Num: the first parameter, byte type, this is output parameter. It is the quantity of DTC.
DTC_Num can be maximum of 8 because of hardware resource limited

DTC[8]: the second parameter, String array type, this is output parameter. It is the String of DTC. The first DTC will be in index 0 of array, The second will be in index 1 of array, ..., Maximum of 8 DTCs

Returns
bool.

5.6.1.51 clearDTC Method

bool clearDTC();

Description

clearDTC will clear all DTCs. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.clearDTC();

Parameters

Nothing

Returns

bool.

5.7 Inner Class GPS

5.7.1 Methods

bool getGPSinfo(float &Latitude,float &Longitude, int &Speed, String &Time, String &Date);
bool getAltitude(float &Altitude);

5.7.1.1 getGPSinfo Method

bool getGPSinfo(float &Latitude ,float &Longitude, int &Speed, String &Time , String &Date);

Description

getGPSinfo will get GPS location and date and time information. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getGPSinfo(float Latitude ,float Longitude, int Speed, String Time , String Date);

Parameters

Latitude: the first parameter, float type, this is output parameter. It is the latitude of vehicle in degree. Plus is north, Minus is south.

Longitude: the 2nd parameter, float type, this is output parameter. It is the Longitude of vehicle in degree. Plus is east, Minus is west.

Speed: the 3rd parameter, int type, this is output parameter. It is the vehicle speed based on GPS navigation in Km/h.

Time: the 4th parameter, String type, this is output parameter. It is the UTC time based on GPS navigation in Format "hh:mm::ss".

Date: the 5th parameter, String type, this is output parameter. It is the UTC date based on GPS navigation in Format "dd/mm/yyyy".

Returns

bool.

5.7.1.2 getAltitude Method

bool getAltitude(float &Altitude);

Description

getAltitude(float &Altitude) will get altitude of vehicle location. It will return true if success, otherwise return false.

Syntax

DFL168A.ISO15765.getAltitude(float Altitude);

Parameters

Altitude: the first parameter, float type, this is output parameter. It is the altitude of vehicle in meter.

Returns

bool.

6 DFL168A asynchronous version library

Asynchronous version library is almost the same as synchronous version except most of methods are unblocked. It will return immediately, not waiting for the finish of DFL168A command.
So for co-operation environment, we should use asynchronous version library.

6.1 Constant

We added 3 new constants for asynchronous version in addition to previous Protocol constants (for 7 old constants, please read Protocol Constant):

- WAITING
- SUCCESS
- FAIL

These constants will be the returned result as un-block method.

When you get WAITING result, you must call this method again later. If you got the SUCCESS or FAIL result, it means that you have done this method absolutely, and then you can call the other method to get vehicle's other parameters.

6.2 Members

Class DFL168A has the following members:

J1939

J1708

ISO15765

GPS

Above members are actually object. We will explain these inner object later. It is almost the same as counter parts of synchronous version

6.3 Methods

DFL168A( );

bool begin(bool intrude,bool Fast);

void end();

byte getOnewireID(byte (&ID)[7]);

byte getDIN(int portNo, bool & Value);

byte setDOUT(int portNo, bool Value);

byte getAnalog(float & ADValue); //0.0 to 999.00

void beginTransparentSerial();

void endTransparentSerial();

bool setExitTransparentKey(byte EndTransparentChar);

bool serialPortAvailable();

bool setSleepDelay(unsigned int SleepDelaySeconds);

Above all methods are the same as one of synchronous version except getOnewireID(byte (&ID)[7]), getDIN(int portNo, bool & Value), setDOUT(int portNo, bool Value),and getAnalog(float & ADValue) .

Let us explain getOnewireID(byte (&ID)[7]) below:

- byte getOnewireID(byte (&ID)[7]);

Description

DFL168A.getOnewireID(byte (&ID)[7]) will get ID from One wire bus device (iButton). It will return SUCCESS if ID succeed to get, otherwise return FAIL if fail in getting ID. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL

Syntax

DFL168A.getOnewireID(ID[7]);

Parameters

ID: the first parameter, 7 bytes array type, this is output parameter. The 7 bytes' ID will be put into this parameter.

Returns

byte

- byte getDIN(int portNo, bool & Value);

Description

getDIN(int portNo, bool & Value) will get input of DFL168A Digital input PortNo. Value True means Logic High, false means Logic Low. It will return SUCCESS if Value succeed to get, otherwise return FAIL if fail in getting Value. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL

Syntax

DFL168A.getDIN(portNo, Value);

Parameters

portNo: the first parameter, int type, this is input parameter. It is DFL168A's Digital input number.
Value: the second parameter, bool type, this is output parameter.

Returns

byte

- byte setDOUT(int portNo, bool Value);

Description

setDOUT(int portNo, bool Value) will set output of DFL168A Digital output PortNo. Value True means Logic High, false means Logic Low. It will return SUCCESS if Value succeed to set, otherwise return FAIL if fail in setting Value. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL

Syntax

DFL168A.setDOUT(portNo,Value);

Parameters

portNo: the first parameter, int type, this is input parameter. It is DFL168A's Digital output number.

Value: the second parameter, bool type, this is input parameter,true means Logic High, false means Logic Low.

Returns

byte

- byte getAnalog(float & ADValue);

Description

getAnalog(float & ADValue) will read analog input of DFL168A. ADValue range: 0.0 to 999.00. It will return SUCCESS if ADValue succeed to get, otherwise return FAIL if fail in getting Value. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL

Syntax

DFL168A.getAnalog(ADValue);

Parameters

ADValue: the first parameter, float type, this is output parameter.

Returns

byte

6.4 Inner Class J1939

6.4.1 Members

Class J1939 has the following members:

PGN65267;

PGN65262;

PGN65256;

PGN65269;

PGN65257;

PGN61444;

PGN61443;

PGN65270;

PGN65271;

PGN65272;

PGN65266;

PGN65263;

PGN65253;

PGN65214;

PGN65248;

PGN65276;

PGN65265;

PGN57344;

PGN64996;

PGN61445;

PGN65268;

Above members are actually object. We will explain these inner object later. They are almost the same as counter parts of synchronous version

6.4.2 Methods

byte getVIN(String &VIN);

byte getDTC(byte &DTC_Num, long (&SPN)[5], byte (&FMI)[5], byte (&CM)[5],byte (&OC)[5], byte DTCFormat=1);

byte clearDTC();

6.4.2.1 getVIN Method

byte getVIN(String &VIN);

Description

J1939.getVIN(String &VIN) will get 19 characters' VIN number from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL

Syntax

DFL168A.J1939.getVIN(VIN);

Parameters

VIN: the first parameter, String type, this is output parameter, It is VIN string of vehicle.

Returns

byte

6.4.2.2 getDTC Method

byte getDTC(byte &DTC_Num, long (&SPN)[5], byte (&FMI)[5], byte (&CM)[5],byte (&OC)[5], byte DTCFormat=1);

Description

This method will get DTC information of vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL. Note: It only can get maximum of 5 DTC because of hardware serial buffer limit

Syntax

DFL168A.J1939.getDTC(DTC_Num,SPN,FMI,CM,OC,DTCFormat);

Parameters

DTC_Num: the first parameter, byte type, this is output parameter, It is quantity of vehicle DTC.

SPN: the second parameter, 5 elements' long array type, this is output parameter, It is SPN number of vehicle.

FMI: the third parameter, 5 elements' byte array type,this is output parameter,It is FMI of vehicle.

CM: the 4th parameter, 5 elements' byte array type,this is output parameter, It is CM of vehicle.

OC: the 5th parameter, 5 elements' byte array type,this is output parameter, It is OC of vehicle.

DTCFormat: the 6th parameter, byte type, this is input parameter,It is DTC Format Version of vehicle. It can be 1, 2, 3, and 4

Returns

byte

6.4.2.3 clearDTC Method

byte clearDTC();

Description

J1939.clearDTC() will clear DTC of vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

Syntax

DFL168A.J1939.clearDTC();

Parameters

Nothing

Returns
byte

6.4.3 Inner Class PGN65267

6.4.3.1 Methods

byte refresh();
bool getLatitude(float &Latitude);
bool getLongitude(float &Longitude);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.3.1.1 refresh Method

byte refresh();

Description

PGN65267.refresh() will refresh PGN65267 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.
If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65267.refresh();

Parameters

Nothing

Returns
byte

6.4.4 Inner Class PGN65262

6.4.4.1 Methods

byte refresh();

bool getCoolantTemperature(int &temp);

bool getFuelTemp(int &temp);

bool getOilTemp(int &temp);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.4.1.1 refresh Method

byte refresh();

Description

PGN65262.refresh() will refresh PGN65262 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65262.refresh();

Parameters

Nothing

Returns

byte

6.4.5 Inner Class PGN65256

6.4.5.1 Methods

byte refresh();

bool getAltitude(float &Altitude);

bool getNavBasedSpeed(float &Speed);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.5.1.1 refresh Method

byte refresh();

Description

PGN65256.refresh() will refresh PGN65256 data from vehicle. It will return SUCCESS if succeed to

get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65256.refresh();

Parameters

Nothing

Returns

byte

6.4.6 Inner Class PGN65269

6.4.6.1 Methods

byte refresh();

bool getBarometricPressure(float &BaroPressure);

bool getAmbientTemp(int &AmbientTemp);

bool getInletTemp(int &InletTemp);

bool getRoadTemp(int &RoadTemp);

bool getCabInteriorTemp(int &CabInteriorTemp);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.6.1.1 refresh Method

byte refresh();

Description

PGN65269.refresh() will refresh PGN65269 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65269.refresh();

Parameters

Nothing

Returns
byte

6.4.7 Inner Class PGN65257

6.4.7.1 Methods

byte refresh();
bool getEngineTripFuel(float &EngineTripFuel);
bool getEngineTotalFuelUsed(float &EngineTotalFuelUsed);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.7.1.1 refresh Method

byte refresh();

Description

PGN65257.refresh() will refresh PGN65257 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.
If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65257.refresh();

Parameters

Nothing

Returns
byte

6.4.8 Inner Class PGN61444

6.4.8.1 Methods

byte refresh();
bool getActualEngineTorque(int &ActualEngineTorque); //-125 to +125 (%)
bool getEngineSpeed(int &EngineSpeed);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.8.1.1 refresh Method

byte refresh();

Description

PGN61444.refresh() will refresh PGN61444 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN61444.refresh();

Parameters

Nothing

Returns

byte

6.4.9 Inner Class PGN61443

6.4.9.1 Methods

byte refresh();

bool getAccelPedalPosi1(float &AccelPedalPosi1);

bool getAccelPedalPosi2(float &AccelPedalPosi2);

bool getEnginePerLoadAtCurrSpeed(int &EnginePerLoadAtCurrSpeed);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.9.1.1 refresh Method

byte refresh();

Description

PGN61443.refresh() will refresh PGN61443 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN61443.refresh();

Parameters

Nothing

Returns
byte

6.4.10 Inner Class PGN65270

6.4.10.1 Methods

byte refresh();
bool getIntakeManifoldPressure(int &IntakeManifoldPressure);
bool getIntakeManifoldTemp(int &IntakeManifoldTemp);
bool getEngineAirInletPressure(int &EngineAirInletPressure);
bool getEngineExhaustGasTemp(int &EngineExhaustGasTemp);
bool getEngineAirFilterDiffPressure(float &EngineAirFilterDiffPressure);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.10.1.1 refresh Method

byte refresh();

Description

PGN65270.refresh() will refresh PGN65270 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65270.refresh();

Parameters

Nothing

Returns
byte

6.4.11 Inner Class PGN65271

6.4.11.1 Methods

byte refresh();

bool getAlternatorVoltage(float &AlternatorVoltage);

bool getElectricalVoltage(float &ElectricalVoltage);

bool getBatteryVoltage(float &BatteryVoltage);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.11.1.1 refresh Method

byte refresh();

Description

PGN65271.refresh() will refresh PGN65271 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65271.refresh();

Parameters

Nothing

Returns

byte

6.4.12 Inner Class PGN65272

6.4.12.1 Methods

byte refresh();

bool getTransmissionOilLevel(float &Percent);

bool getTransmissionOilLevelHighLow(float &HighLow);

bool getTransmissionOilPressure(float &Pressure);

bool getTransmissionOilTemp(float &Temperature);.

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.12.1.1 refresh Method

byte refresh();

Description

PGN65272.refresh() will refresh PGN65272 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65272.refresh();

Parameters

Nothing

Returns

byte

6.4.13 Inner Class PGN65266

6.4.13.1 Methods

byte refresh();

bool getFuelRate(float &FuelRate);

bool getInstantFuelEconomy(float &InstantFuelEconomy);

bool getAvgFuelEconomy(float &AvgFuelEconomy);

bool getEngineThrottlePos(float &EngineThrottlePos);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.13.1.1 refresh Method

byte refresh();

Description

PGN65266.refresh() will refresh PGN65266 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65266.refresh();

Parameters

Nothing

Returns
byte

6.4.14 Inner Class PGN65263

6.4.14.1 Methods

byte refresh();

bool getFueDeliveryPressure(int &FueDeliveryPressure);

bool getEngineOilLevel(float &EngineOilLevel);

bool getEngineOilPressure(int &EngineOilPressure);

bool getEngineCoolantPressure(int &EngineCoolantPressure);

bool getEngineCoolantLevel(float &EngineCoolantLevel);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.14.1.1 refresh Method

byte refresh();

Description

PGN65263.refresh() will refresh PGN65263 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65263.refresh();

Parameters

Nothing

Returns
byte

6.4.15 Inner Class PGN65253

6.4.15.1 Methods

byte refresh();

bool getTotalEngineHours(float &TotalEngineHours);

bool getTotalEngineRevolutions(float &TotalEngineRevolutions);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.15.1.1 refresh Method

byte refresh();

Description

PGN65253.refresh() will refresh PGN65253 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65253.refresh();

Parameters

Nothing

Returns

byte

6.4.16 Inner Class PGN65214

6.4.16.1 Methods

byte refresh();

bool getRatedEngineSpeed(float &RatedEngineSpeed);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.16.1.1 refresh Method

byte refresh();

Description

PGN65214.refresh() will refresh PGN65214 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65214.refresh();

Parameters

Nothing

Returns
byte

6.4.17 Inner Class PGN65248

6.4.17.1 Methods

byte refresh();

bool getTripDistance(float &TripDistance);

bool getTotalDistance(float &TotalDistance);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.17.1.1 refresh Method

byte refresh();

Description

PGN65248.refresh() will refresh PGN65248 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65248.refresh();

Parameters

Nothing

Returns
byte

6.4.18 Inner Class PGN65276

6.4.18.1 Methods

byte refresh();

bool getWasherFluidLevel(float &WasherFluidLevel);

bool getFuelLevel1(float &FuelLevel1);

bool getFuelLevel2(float &FuelLevel2);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.18.1.1 refresh Method

byte refresh();

Description

PGN65276.refresh() will refresh PGN65276 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65276.refresh();

Parameters

Nothing

Returns

byte

6.4.19 Inner Class PGN65265

6.4.19.1 Methods

byte refresh();
bool getWheelBasedVehicleSpeed(float &WheelBasedVehicleSpeed);

bool getParkingBrake(bool &ParkingBrakeSet);

bool getBrake(bool &BrakePedalDepressed);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.19.1.1 refresh Method

byte refresh();

Description

PGN65265.refresh() will refresh PGN65265 data from vehicle.It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65265.refresh();

Parameters

Nothing

Returns
byte

6.4.20 Inner Class PGN57344

6.4.20.1 Methods

byte refresh();
bool getSeatBelt(bool buckled);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.20.1.1 refresh Method

byte refresh();

Description

PGN57344.refresh() will refresh PGN57344 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN57344.refresh();

Parameters

Nothing

Returns
byte

6.4.21 Inner Class PGN64996

6.4.21.1 Methods

byte refresh();
bool getPayLoad(int &PayLoad);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.21.1.1 refresh Method

byte refresh();

Description

PGN64996.refresh() will refresh PGN64996 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN64996.refresh();

Parameters

Nothing

Returns

byte

6.4.22 Inner Class PGN61445

6.4.22.1 Methods

byte refresh();

bool getCurrentGear(int &CurrentGear);

bool getSelectedGear(int &SelectedGear);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.22.1.1 refresh Method

byte refresh();

Description

PGN61445.refresh() will refresh PGN61445 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN61445.refresh();

Parameters

Nothing

Returns
byte

6.4.23 Inner Class PGN65268

6.4.23.1 Methods

byte refresh();

bool getTirePressure(int &TirePressure);

bool getTireTemperature(float &Temperature);

bool getTireLocation(int &Front2RearNumber, int &Left2RighNumber);

bool getTireValvePressureMonitor(int &TireValvePressureMonitor);

Only method refresh() is different from synchronous version. So we only explain method refresh().

6.4.23.1.1 refresh Method

byte refresh();

Description

PGN65268.refresh() will refresh PGN65268 data from vehicle. It will return SUCCESS if succeed to get, otherwise return FAIL if fail in getting. It will return WAITING if dfl168A need more time, and you will call this method again later on. You cannot call the other method before you get SUCCESS or FAIL.

If you want to get latest vehicle data from the other methods in this PGN, you should call this method firstly.

Syntax

DFL168A.J1939.PGN65268.refresh();

Parameters

Nothing

Returns
byte

6.5 Inner Class J1708

6.5.1 Methods

byte getAirPressure(int &AirPressure);
byte getEngineOilPressure(int &EngineOilPressure);
byte getEngineCoolantPressure(int &EngineCoolantPressure);
byte getFuelLevel1(float &FuelLevel1);
byte getFuelLevel2(float &FuelLevel2);
byte getBarometricPressure(float &Pressure);
byte getEngineThrottlePos(float &EngineThrottlePos);
byte getWasherFluidLevel(float &WasherFluidLevel);
byte getVehicleSpeed(float &VehicleSpeed);
byte getAccelPedalPosi1(float &AccelPedalPosi1);
byte getAccelPedalPosi2(float &AccelPedalPosi2);
byte getAccelPedalPosi3(float &AccelPedalPosi3);
byte getEngineLoad(float &Percent);
byte getEngineOilLevel(float &EngineOilLevel);
byte getCoolantTemperature(int &temp);
byte getEngineCoolantLevel(float &EngineCoolantLevel);
byte getTransmissionOilLevel(float &Percent);
byte getTransmissionOilLevelHighLow(float &HighLow);
byte getTransmissionOilPressure(float &Pressure);
byte getTransmissionOilTemp(float &Temperature);
byte getPowerSpecificInstantFuelEconomy(float &Rate);
byte getAvgFuelRate(float &FuelRate);
byte getInstantFuelEconomy(float &InstantFuelEconomy);
byte getAvgFuelEconomy(float &AvgFuelEconomy);
byte getElectricalVoltage(float &BatteryVoltage);
byte getRatedEnginePower(float &Power);
byte getBatteryVoltage(float &BatteryVoltage);
byte getAlternatorVoltage(float &AlternatorVoltage);
byte getAmbientTemp(int &AmbientTemp);
byte getCargoAmbientTemp(int &CargoTemp);
byte getRoadTemp(int &RoadTemp);
byte getCabInteriorTemp(int &CabInteriorTemp);
byte getInletTemp(int &InletTemp);
byte getFuelTemp(int &temp);
byte getOilTemp(int &temp);
byte getCargoWeight(float &CargoW);
byte getEngineTripFuel(float &EngineTripFuel);

byte getEngineTotalFuelUsed(float &EngineTotalFuelUsed);
byte getFuelRate(float &FuelRate);
byte getRatedEngineSpeed(float &RatedEngineSpeed);
byte getEngineSpeed(int &EngineSpeed);
byte getIntakeManifoldTemp(int &IntakeManifoldTemp);
byte getPowerTakeoffStatus(bool &PTOModeActive, bool &ClutchSwitchOn, bool &BrakeSwitchOn,
bool &AccelSwitchOn, bool &ResumeSwitchOn, bool &CoastSwitchOn,
bool &SetSwitchOn, bool &PTOControlSwitchOn);
byte getTripDistance(float &TripDistance);
byte getTotalDistance(float &TotalDistance);
byte getTotalEngineHours(float &TotalEngineHours);
byte getTotalEngineRevolutions(float &TotalEngineRevolutions);
byte getVIN(String &VIN);
byte getDTC(byte &DTC_Num,byte &MID,int (&PID_SID)[8],bool (&IsPID)[8], byte (&FMI)[8],
bool (& IsActive)[8],bool (&OccurrenceExist)[8],byte (&OccurrenceCount)[8]);
byte clearDTC(byte MID,int PID_SID, bool IsPID);
byte getFaultDescription(byte MID,int PID_SID, bool IsPID,byte FMI, String & FaultDescription);
byte getPIDSIDDescription(byte MID,int PID_SID, bool IsPID, String & PID_SID_Description);

All methods are almost the same as counter parts of synchronous version except returned value byte instead of bool.

Returned value can be one of "SUCCESS","FAIL", and "WAITING".
If returned value is "WAITING", you must call the same methods again later on. In this situation, you cannot call the other method.
When you got returned value of "SUCCESS" or "FAIL", you can call the other method.

6.6 Inner Class ISO15765

6.6.1 Methods

byte getCoolantTemperature(int &temp);
byte getEngineSpeed(int &EngineSpeed);
byte getVehicleSpeed(float &VehicleSpeed);
byte getIntakeManifoldPressure(int &IntakeManifoldPressure);
byte getFuelSystemStatus(bool &A_Openloop, bool &A_Closedloop,
bool &A_OpenloopByDriving_Con, bool &A_OpenloopByFault,
bool &A_ClosedloopButFault, bool &B_Openloop,

```
                                bool &B_Closedloop,      bool &B_OpenloopByDriving_Con,
                                bool &B_OpenloopByFault,   bool &B_ClosedloopButFault);
byte getCalculatedLoadValue(int &CalculatedLoad);
byte getShortTermFuelTrimBank13(float &Bank1,float &Bank3);
byte getLongTermFuelTrimBank13(float &Bank1,float &Bank3);
byte getShortTermFuelTrimBank24(float &Bank2,float &Bank4);
byte getLongTermFuelTrimBank24(float &Bank2,float &Bank4);
byte getIgnitionTimingAdvance(float &Angle);
byte getIntakeAirTemperature(int &temp);
byte getAirFlowRateFrmMAF(float &FlowRate);
byte getAbsThrottlePosition(float &Percent);
byte getOxygenSensorLocation(bool &Bank1_Sensor1Present,  bool &Bank1_Sensor2Present,
                             bool &Bank1_Sensor3Present,  bool &Bank1_Sensor4Present,
                             bool &Bank3_Sensor1Present,  bool &Bank3_Sensor2Present,
                             bool &Bank2_Sensor1Present,  bool &Bank2_Sensor2Present,
                             bool &Bank2_Sensor3Present,  bool &Bank2_Sensor4Present,
                             bool &Bank4_Sensor1Present,   bool &Bank4_Sensor2Present);
byte getBank1OSensor1Voltage(float &OutVoltage);
byte getBank1OSensor2Voltage(float &OutVoltage);
byte getBank1OSensor3Voltage(float &OutVoltage);
byte getBank1OSensor4Voltage(float &OutVoltage);
byte getBank2OSensor1Voltage(float &OutVoltage);
byte getBank2OSensor2Voltage(float &OutVoltage);
byte getBank2OSensor3Voltage(float &OutVoltage);
byte getBank2OSensor4Voltage(float &OutVoltage);
byte getBank3OSensor1Voltage(float &OutVoltage);
byte getBank3OSensor2Voltage(float &OutVoltage);
byte getBank4OSensor1Voltage(float &OutVoltage);
byte getBank4OSensor2Voltage(float &OutVoltage);
byte getOBDCertified(String &OBD);
byte getTimeSinceEngineStart(unsigned int &TotalTime);
byte getDistanceTraveledMIL(unsigned int &Distance);
byte getFuelRailPressure(float &Pressure);
byte getFuelLevelInput(float &Percent);
byte getDistanceTraveledSinceDTC_Clear(unsigned int &Distance);
byte getBarometricPressure(int &Pressure);
byte getControlModuleVoltage(float &Voltage);
byte getRelativeThrottlePosition(float &Percent);
byte getAmbientTemp(int &AmbientTemp);
```

```
byte getCommandedThrottleActuatorControl(float &Percent);
byte getEngineRunTimeMIL(unsigned int &TotalTime);
byte getEngineRunTimeSinceDTC_Clear(unsigned int &TotalTime);
byte getTypeOfFuelUsedCurrently(String & FuelType);
byte getRelativeAcceleratorPedalPosition(float &Percent);
byte getHybridBatteryPackRemainingLife(float &Percent);
byte getEngineOilTemperature(int &Tem);
byte getFuelRate(float &FuelRate);
byte getActualEngineTorque(int &ActualEngineTorque);
byte getMILStatus(bool &MIL_IS_ON);
byte getEngineRunTime(unsigned long &TotalEngineRunTime, unsigned long &TotalIdleRunTime,
                      unsigned long &TotalRunTimeWithPTOActive);
byte getVIN(String &VIN);
byte getDTC(byte &DTC_Num,String ( &DTC)[8]);
byte clearDTC();
```

All methods are almost the same as counter parts of synchronous version except returned value byte instead of bool.

Returned value can be one of "SUCCESS","FAIL", and "WAITING".
If returned value is "WAITING", you must call the same methods again later on. In this situation, you cannot call the other method.
When you got returned value of "SUCCESS" or "FAIL", you can call the other method.

6.7 Inner Class GPS

6.7.1 Methods

byte getGPSinfo(float &Latitude,float &Longitude, int &Speed, String &Time, String &Date);
byte getAltitude(float &Altitude);

All methods are almost the same as counter parts of synchronous version except returned value byte instead of bool.

Returned value can be one of "SUCCESS","FAIL", and "WAITING".
If returned value is "WAITING", you must call the same methods again later on. In this situation, you cannot call the other method.
When you got returned value of "SUCCESS" or "FAIL", you can call the other method.

7 Examples

All examples for synchronous version is in "Your Source code Location\DFL168A_Sync_examples"

All examples for asynchronous version is in "Your Source code Location\DFL168A_Async_examples"