DFL168A Python Module

Table of Contents

1 Features Highlights

- Simply import DFL168A module and use function 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.
- Only Synchronous version can be used. Please use DFL168A in one thread or one process, your other application in the other thread or process.
- Can be used in different platforms such as Windows, Linux, MacOS
- Python 3.X version

2 How to install library?

Install Library is very simple:

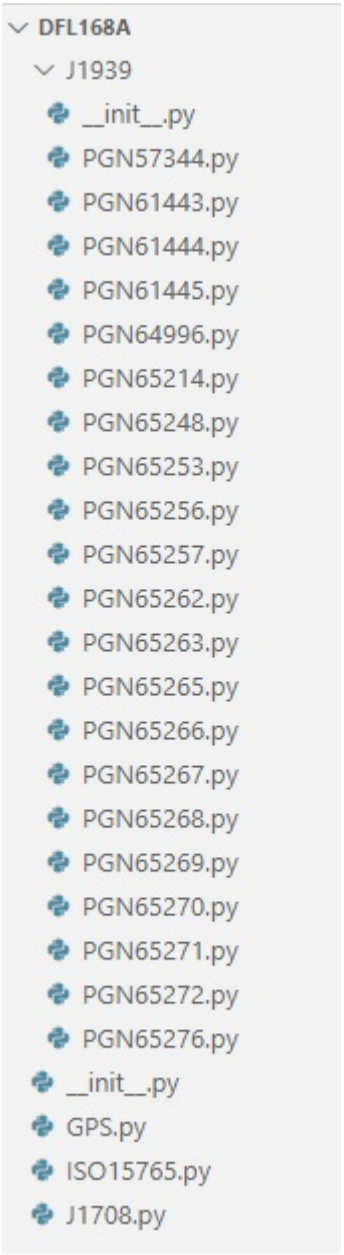
You just use command below:

pip install pyserial (If you have no installed pyserial module)

pip install DFL168A

3 How to use Library?

DFL168A Library structure is shown blow:



We can know that we have modules below:

DFL168A module

DFL168A.ISO15765 module

DFL168A.J1708 module

DFL168A.GPS module

DFL168A.J1939 module

DFL1689.J1939.PGN57344 module

DFL1689.J1939.PGN61443 module

.....

DFL1689.J1939.PGN65272 module

DFL1689.J1939.PGN65276 module

Every module provides lots of function for customer.

The most important function is "begin" function in DFL168A module. begin function definition is below:

r, reason =DFL168A.begin(SerialName, BaudRate=57600,Protocol=PROTOCOL.AUTO,BaudRate4Protocol=250000,Timeout=0.5,BaudRate4GPS=9600,TimeOut4GPS=0.5,Intrude=True,Fast=False)

Arguments: SerialName is serial port name, it is string 'COM1' or 'COM2', and so on' in windows, it is string '/dev/ttyS0' , or '/dev/ttyS1', and so on. If you use USB/Uart adaptor in linux, SerialName is string '/dev/ttyUSB0' , or '/dev/ttyUSB1', and so on.

The other arguments are easy to understand from their name. We will explain in DFLA68A module

When we call this function, software will open a serial port, and set up necessary parameter for DFL168A, and verify vehicle protocol. This function will run for around 5 seconds

Function return 2 values: r and reason. r is bool value. True denotes DFL168A OK. False denotes DFL168A has something wrong. reason denotes specific reason. It is enumeration constant. It has value "DFL168A.REASON.NORMAL", "DFL168A.REASON.FAIL_IN_OPEN_SERIAL", "DFL168A.REASON.FAIL_IN_AT_CMD", "DFL168A.REASON.FAIL_IN_WRONG_IC", "DFL168A.REASON.SLEEP_WARN"

The opposite function will be "end" function. It will close Serial Port which is open by begin function.

We will give some example for how to use DFL168A module. These examples are only for DFL168A. You should put all DFL168A code into another thread or process because all DFL168A function is block function.

Lets give example for J1708/J1587:

The python code is shown below:

```
import DFL168A, keyboard
from DFL168A import J1708
x,Y=DFL168A.begin('COM4',57600,DFL168A.PROTOCOL.J1708,250000,1.0)
print("Reason:")
print(Y)
if x and Y!=DFL168A.REASON.SLEEP_WARN:
    while True:
        if keyboard.is_pressed('q'):
            break
        print("Sleep Warning True?:")
        print (str(DFL168A.SleepWarning))
      r,VIN=J1708.getVIN()
      if r:
          print(VIN)
```

```
15          else:
16              print('fail in VIN')
17          r, DTC_Num,MID,PID_SID,IsPID,FMI,IsActive,OccurrenceExist,OccurrenceCount=J1708.getDTC()
18          if r:
19              print("DTC Num: ")
20              print(DTC_Num)
21              print("MID: ")
22              print(MID)
23              print("PID_SID: ")
24              print(PID_SID)
25              print("IsPID: ")
26              print(IsPID)
27              print("FMI: ")
28              print(FMI)
29              print("IsActive: ")
30              print(IsActive)
31              print("OccurrenceExist: ")
32              print(OccurrenceExist)
33              print("OccurrenceCount: ")
34              print(OccurrenceCount)
35          else:
36              print('fail in getting DTC')
37          r,VehicleSpeed=J1708.getVehicleSpeed()
38          if r:
39            print("Vehicle Speed: %8.3f" % VehicleSpeed)
40      DFL168A.End()
```

Line 3 statement will setup DFL168A IC which may run for around 5 seconds. Line 6 statement will tell us code will run the next statements if DFL168A IC is OK and without sleep warning, otherwise, code run line 40 statement to close serial port and end this program.

line 7 to line 39 are forever loop. It exit this loop when we press key "q". In this forever loop, code will do normal vehicle motor data detection. Line 11 will print DFL168A.SleepWarning variable value. If DFL168A.SleepWarning is true, it means that Vehicle Engine is shut down or Vehicle Communication wiring is broken (DFL168A didn't get vehicle data for at least 5 seconds). You can change line 11 statement to let code do your expected action. Line 12 statement is for getting vehicle identification number (VIN). Line 17 statement is for getting DTC information. Line 37 statement is for getting vehicle speed.

Lets give example for J1939.

The python code is shown below:

```
1     import DFL168A, keyboard
2     from DFL168A import J1939
3     from DFL168A.J1939 import PGN61444
4     x,Y=DFL168A.begin('COM4',57600,DFL168A.PROTOCOL.J1939,250000,1.0)
5     print("Reason:")
6     print(Y)
7     if x and Y!=DFL168A.REASON.SLEEP_WARN:
8         while True:
9           if keyboard.is_pressed('q'):
10              break
11            print("Sleep Warning True?:")
```

```
12            print (str(DFL168A.SleepWarning))
13            r,VIN=J1939.getVin()
14            if r:
15                print(VIN)
16            else:
17                print('fail in VIN')
18            r, DTC_Num,SPN,FMI,CM,OC=J1939.getDTC(DTCFormat=4)
19            if r:
20                print("DTC Num: \r\n")
21                print(DTC_Num)
22            else:
23                print('fail in getting DTC')
24            r=J1939.clearDTC()
25            if r:
26                print('Clear DTC sucess \r\n')
27            else:
28                print('fail in clear DTC\r\n')
29            r=PGN61444.refresh()
30            if r:
31                print("PGN61444 Success")
32                r,lat=PGN61444.getActualEngineTorque()
33                if r:
34                    print('aCtual EngineTorq is %f \r\n'% lat)
25                else:
36                    print('aCtual EngineTorq wrong \r\n')
37                r,lat=PGN61444.getEngineSpeed()
38                if r:
39                    print('Engine speed is %f \r\n'% lat)
40                else:
41                    print('Engine speed wrong \r\n')
42            else:
43                print("PGN61444 Fail")
44    DFL168A.End()
```

J1939 module is different from the other protocol module. For getting VIN, DTC , and Clearing DTC, you can use J1939 module function directly. However, for the other parameters, you must use PGNxxxxx (xxxxx denotes PGN number such as 61444) sub-module. Firstly, you should call PNGxxxxx module's refresh function. Then you can call other functions to get parameter under this PGN only when refesh function returns True value.

Line 4 statement will setup DFL168A IC which may run for around 5 seconds. Line 7 statement will tell us code will run the next statements if DFL168A IC is OK and without sleep warning, otherwise, code run line 44 statement to close serial port and end this program.

line 8 to line 43 are forever loop. It exit this loop when we press key "q". In this forever loop, code will do normal vehicle motor data detection. Line 12 will print DFL168A.SleepWarning variable value. If DFL168A.SleepWarning is true, it means that Vehicle Engine is shut down or Vehicle Communication wiring is broken (DFL168A didn't get vehicle data for at least 5 seconds). You can change line 12 statement to let code do your expected action. Line 13 statement is for getting vehicle identification number (VIN). Line 18 statement is for getting DTC information. Line 24 statement is for Reset Fault code (Clear DTC). Line 29 statement will update PGN 61444 data. Line 30 statement will decide whether PGN61444 updating is successful. If it is successful, code will run Line 31 to Line 41. Line 32 is for getting actual Engine Torque. Line 37 is for getting engine speed.

Lets give example for ISO15765.

The python code is shown below:

```
1   import DFL168A, keyboard
2   from DFL168A import ISO15765
3   x,Y=DFL168A.begin('COM4',57600,DFL168A.PROTOCOL.AUTO)
4   print("Reason:")
5   print(Y)
6   if x and Y!=DFL168A.REASON.SLEEP_WARN:
7       while True:
8           if keyboard.is_pressed('q'):
9               break
10          print("Sleep Warning True?:")
11          print (str(DFL168A.SleepWarning))
12          r,VIN=ISO15765.getVIN()
13          if r:
14              print(VIN)
15          else:
16              print('fail in VIN')
17          r, DTC_Num,DTC=ISO15765.getDTC()
18          if r:
19              print("DTC Num: ")
20              print(DTC_Num)
21              print("DTC: ")
22              print(DTC)
23          else:
24              print('fail in getting DTC')
25          r,VehicleSpeed=ISO15765.getVehicleSpeed()
26          if r:
27              print("Vehicle Speed: %8.3f" % VehicleSpeed)
28          r=ISO15765.clearDTC()
29          if r:
30              print('Clear DTC sucess \r\n')
31          else:
32              print('fail in clear DTC\r\n')
33  DFL168A.End()
```

Line 3 statement will setup DFL168A IC which may run for around 5 seconds. Line 6 statement will tell us code will run the next statements if DFL168A IC is OK and without sleep warning, otherwise, code run line 33 statement to close serial port and end this program.

line 7 to line 32 are forever loop. It exit this loop when we press key "q". In this forever loop, code will do normal vehicle motor data detection. Line 11 will print DFL168A.SleepWarning variable value. If DFL168A.SleepWarning is true, it means that Vehicle Engine is shut down or Vehicle Communication wiring is broken (DFL168A didn't get vehicle data for at least 5 seconds). You can change line 11 statement to let code do your expected action. Line 12 statement is for getting vehicle identification number (VIN). Line 17 statement is for getting DTC information. Line 25 statement is for getting vehicle speed. Line 28 statement is for Reset Fault code (Clear DTC)

Lets give example for GPS.

The python code is shown below:

```
1    import DFL168A, keyboard
2    from DFL168A import GPS
3    x,Y=DFL168A.begin('COM4',57600,DFL168A.PROTOCOL.J1708,250000,0.5,9600,1.0)
4    print("Reason:")
5    print(Y)
6    if x and Y!=DFL168A.REASON.SLEEP_WARN:
7        while True:
8            if keyboard.is_pressed('q'):
9                break
10           print("Sleep Warning True?:")
11           print (str(DFL168A.SleepWarning))
12           r,Latitude,Longitude,Speed, Time,Date=GPS.getGPSinfo()
13           if r:
14               print("Latitude: ")
15               print(Latitude)
16               print("Longitude: ")
17               print(Longitude)
18               print("Speed: ")
19               print(Speed)
20               print("Time: ")
21               print(Time)
22               print("Date: ")
23               print(Date)
24           else:
25               print('fail in getting GPS information')
26           r, Altitude=GPS.getAltitude()
27           if r:
28               print("Altitude: ")
29               print(Altitude)
30           else:
31              print('fail in getting Altitude')
32   DFL168A.End()
```

Line 3 statement will setup DFL168A IC (GPS baud rate is 9600, GPS timeout is 1.0 second) , which may run for around 5 seconds. Line 6 statement will tell us code will run the next statements if DFL168A IC is OK and without sleep warning, otherwise, code run line 32 statement to close serial port and end this program.

line 7 to line 31 are forever loop. It exit this loop when we press key "q". In this forever loop, code will do normal GPS data detection (Of cause, you can do normal vehicle motor data detection too). Line 11 will print DFL168A.SleepWarning variable value. If DFL168A.SleepWarning is true, it means that Vehicle Engine is shut down or Vehicle Communication wiring is broken (DFL168A didn't get vehicle data for at least 5 seconds). You can change line 11 statement to let code do your expected action. Line 12 statement is for getting GPS information. Line 26 statement is for getting altitude information.

Lets give example for other items such as i-button, digital input/output, analog input, transparent main uart to access device1 (uart2).

The python code is shown below:

```
1    import DFL168A, keyboard
2    x,Y=DFL168A.begin('COM4',57600,DFL168A.PROTOCOL.J1708,250000,0.5,9600,1.0)
3    print("Reason:")
4    print(Y)
5    if x and Y!=DFL168A.REASON.SLEEP_WARN:
6        if DFL168A.setSleepDelay(3600):   #delay 1 hour for sleep
7            print('DFL168A Sleep delay change to 1 hour')
8        while True:
9            if keyboard.is_pressed('q'):
10               break
11           print("Sleep Warning True?:")
12           print (str(DFL168A.SleepWarning))
13           r,I_button_ID=DFL168A.getOnewireID()
14           if r:
15               print("I button ID: ")
16               print(I_button_ID)
17           else:
18               print('fail in getting I button ID')
19           r, din0=DFL168A.getDIN(0)
20           if r:
21               print("Din0: ")
22               if din0: print("High")
23               else: print("Low")
24           else:
25               print('fail in getting Din0')
26           r, Ain=DFL168A.getAnalog()
27           if r:
28               print("Analog input: ")
29               print(Ain)
30           else:
31               print('fail in getting Analog input')
32           r=DFL168A.setDOUT(0,False)
33           if r:
34               print("Set Dout0 to Low successfully ")
35           else:
36               print('fail in seting Dout0')
37           r=DFL168A.setExitTransparentKey(ord('A'))  #if default exit charactor, remove this statement
38           if r:
39               DFL168A.beginTransparentSerial()
40               print("Starting access DEV1 transparently ")
41               DFL168A.Ser4DFL168A.write('Hello DFL168A'.encode('utf-8'))
42               DFL168A.endTransparentSerial()
43               if not DFL168A.TransparentSerialAvailable:
44                   print('DFL168A end transparent mode, and resume to normal mode')
45               else:
46                   print('Fail in ending transparent mode')
49                   break
50   DFL168A.End()
```

Line 2 statement will setup DFL168A IC , which may run for around 5 seconds. Line 5 statement will tell us code will run the next statements if DFL168A IC is OK and without sleep warning, otherwise, code run line 50 statement to close serial port and end this program.

Line 6 will set up sleep delay time to 1 hour.

Line 8 to line 49 are forever loop. It exit this loop when we press key "q" or fail in quitting to transparent mode . In this forever loop, code will do normal GPIO or I-button operation (Of cause, you can do normal

vehicle motor data detection too). Line 12 will print DFL168A.SleepWarning variable value. If DFL168A.SleepWarning is true, it means that Vehicle Engine is shut down or Vehicle Communication wiring is broken (DFL168A didn't get vehicle data for at least 5 seconds). You can change line 12 statement to let code do your expected action. Line 13 statement is for getting I-button ID. Line 19 statement is for getting Din0 value. Line 26 statement is for getting analog input value. Line 32 statement will set Dout0 to Logic low. Line 37 statement set up exiting transparent character to 'A'. Line 39 statement will enter transparant mode, so our Python serial port can access DEV1 (DFL168A UART2) . In line 41, we send message "Hello DFL168A" to DEV1 . Line 42 statement will exit "transparent mode" and return to normal working mode.

4 DFL168A Module

DFL168A module has sub-module J1939, J1708,ISO15765,and GPS, J1939 sub-module has lots of PGN sub modules such as PGN65267,..., PGN57344.

For J1939, if you want some vehicle data, you should know which PGN it belongs to. And firstly, you should call its refresh function of PGN sub-module. After that, you can use the PGN sub-module's other functions to get vehicle data. All functions are blocked, so all functions will be running and exit only when the function finish its job.

For J1708 sub-module, there are lots of functions which will give you vehicle data. These functions are blocked.

For ISO15765 sub-module, there are lots of functions which will give you vehicle data. These functions are blocked.

For GPS sub-module, there are functions which will give you GPS data. These functions are blocked.

4.1 Constant

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

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

PROTOCOL.ISO15765_11_500----------This is ISO15765 protocol with 11 bits CAN ID and 500Kbps baud rate.

PROTOCOL.ISO15765_29_500----------This is ISO15765 protocol with 29 bits CAN ID and 500Kbps baud rate.

PROTOCOL.ISO15765_11_250----------This is ISO15765 protocol with 11 bits CAN ID and 250Kbps baud rate.

PROTOCOL.ISO15765_29_250----------This is ISO15765 protocol with 29 bits CAN ID and 250Kbps baud rate.

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

PROTOCOL.J1708----------This is J1708/J1587 protocol,

For IC failure reason, we have the following enumeration constant in the DFL168A Module:

REASON.NORMAL----------DFL168A has no any problem when we call begin function to set up IC

REASON.FAIL_IN_OPEN_SERIAL----------Open specified serial port fail when we call begin function to set up IC

REASON.FAIL_IN_AT_CMD----------DFL168A cannot execute AT Command when we call begin function to set up IC, please check wiring between DFL168A Serial port and Python computer serial port

REASON.FAIL_IN_WRONG_IC----------IC is not DFL168A when we call "begin function" to set up IC, please check whether we soldered correct DFL168A IC on board

REASON.SLEEP_WARN----------DFL168A got sleep warning information when we call begin function to set up IC,please check whether vehicle protocol is correct or wiring is ok between DFL168A and vehicle

4.2 global variables

There is important global variables for customers:

SleepWarning: This is bool value. It denotes Engine Off or wrong protocol or bad wiring between DFL168A and vehicle when its value is "True". This variable value changes automatically. DFL168A will sleep after defined time delay [default dealy is 175 Seconds or around 2.92 minutes, you can use function setSleepDelay (Delay_seconds) to change it]. We can use this variable to know whether Engine is off/on if wiring and protocol are OK. And we can let host board be shut down safely (Linux or Windows or Mac OS) when engine is off.

TransparentSerialAvailable: This is bool value. It denotes that we can access DEV1 (UART2) directly by Ser4DFL168A object when its value is "True"

Ser4DFL168A: This is Serial object (From pyserial module) . When TransparentSerialAvailable is true (Set by function beginTransparentSerial()), we can use Ser4DFL168A object write and read method to access DFL168A Uart2 directly.

4.3 Functions

DFL168A module has the following functions:

begin
getOnewireID
getDIN
setDOUT
getAnalog
setExitTransparentKey
beginTransparentSerial
endTransparentSerial
setSleepDelay
End

We will explain these functions.

4.3.1 begin Function

begin(SerialName, BaudRate=57600,Protocol=PROTOCOL.AUTO,BaudRate4Protocol=250000, Timeout=0.5,BaudRate4GPS=9600,TimeOut4GPS=0.5,Intrude=True,Fast=False)

Description

DFL168A.begin function will open a serial port in host and config DFL168A IC and check whether DFL168A Configuration is OK. If configuration succeed, it will return true, otherwise return false. This method will run as long as 5 seconds for setting DFL168A IC.

Syntax

r, Reason = DFL168A.begin(SerialName, BaudRate=57600,Protocol=PROTOCOL.AUTO, BaudRate4Protocol=250000,Timeout=0.5,BaudRate4GPS=9600,TimeOut4GPS=0.5,Intrude=True, Fast=False)

Parameters

SerialName: the first parameter, string type, this is input parameter. It denotes Host Serial port name. In windows, it is "COM1", "COM2",..., and so on. In Linux, it is "/dev/ttyS0", "/dev/ttyS1", ..., and so on. Or "/dev/ttyUSB0", "/dev/ttyUSB1", ..., and so on.

BaudRate: the 2nd parameter, int type, this is input parameter. it denotes DFL168A Uart1 baud

rate, default value is 57600.

Protocol: the 3rd parameter, enumeration PROTOCOL type, this is input parameter. it denotes what protocol we used. It should be one of "PROTOCOL.AUTO", "PROTOCOL. ISO15765_11_500","PROTOCOL.ISO15765_29_500","PROTOCOL.ISO15765_11_250", "PROTOCOL.ISO15765_29_250", "PROTOCOL.J1939", "PROTOCOL.J1708"

BaudRate4Protocol: the 4th parameter, int type, this is input parameter. it denotes J1939 Protocol baud rate, default value is 250000

Timeout: the 5th parameter, float type, this is input parameter. it denotes vehicle time out value in second unit, default value is 0.5 seconds.

BaudRate4GPS: the 6th parameter, int type, this is input parameter. it denotes GPS baud rate (UART2 baud rate), default value is 9600

TimeOut4GPS: the 7th parameter, float type, this is input parameter. it denotes GPS command time out value in second unit, default value is 0.5 seconds. In general , it is GPS NMEA sentence broadcast periods.

Intrude: the 8th 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 9th 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. Module will send all config data to DFL168A if its value is false. You don't need take care of anything. Module takes care of everything for you.

Returns

r : bool, Ture denotes DFL168A Configuration is OK (But vehicle protocol may be wrong, you should identify by "Reason" value)

Reason: enumeration REASON type. It should be one of "REASON.NORMAL","REASON. FAIL_IN_OPEN_SERIAL","REASON.FAIL_IN_AT_CMD", "REASON.FAIL_IN_WRONG_IC", and "REASON.SLEEP_WARN".

We can use this function to choose correct vehicle protocol automatically. For ISO15765, it has no problem, we just use "PROTOCOL.AUTO" for calling begin function. When we use "PROTOCOL. AUTO" to call begin function, DFL168A will select one of 4 ISO15765 protocols automatically. If we got correct protocol, return r must be "True" and Reason must be "REASON.NORMAL". If Reason is "REASON.SLEEP_WARN", we should call end() function, and then call begin() function with "PROTOCOL.J1939". If we got correct protocol, return r must be "True" and Reason must be "REASON.NORMAL". If Reason is "REASON.SLEEP_WARN", we should call end() function, and then call begin() function with "PROTOCOL.J1708".

4.3.2 end Function

end()

Description

DFL168A.end() will close Python serial port which is used by DFL168A, and You can use DFL168A.begin function again only after you run this function.

Syntax

DFL168A.end()

Parameters

Nothing

Returns

Nothing

4.3.3 getOnewireID Function

getOnewireID( )

Description

DFL168A.getOnewireID( ) will get ID from One wire bus device (iButton). It will return true if ID succeed to get, otherwise return false. ID is second return value as list

Syntax

r, t = DFL168A.getOnewireID( )

Parameters

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

Returns

r : bool, Ture denotes I-button is OK

t: list, it contains 7 int type ID.

4.3.4 getDIN Function

getDIN(portNo);

Description

getDIN(portNo) will return Digital input of DFL168A Digital Input. The second returned value True means Logic High, false means Logic Low.

Syntax

r,t = DFL168A.getDIN(portNo);

Parameters

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

Returns

r : bool, Ture denotes we get digital input successfully.

t: bool , True denote digital input is Logic High, otherwise it is Logic Low.

4.3.5 setDOUT Function

setDOUT(portNo,Value)

Description

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

Syntax

r = 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

r : bool, Ture denotes we set digital output successfully.

4.3.6 getAnalog Function

getAnalog()

Description

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

Syntax

r, t = DFL168A.getAnalog()

Parameters

Nothing

Returns

r : bool, Ture denotes we get digital input successfully.

t: float , It is analog input value, range: 0.0 to 999.00

4.3.7 beginTransparentSerial Function

beginTransparentSerial()

Description

beginTransparentSerial() will use hardware Serial of Python to access DEV1 (Another serial port: Uart2) of DFL168A directly. This hardware Serial was used for DFL168A Command in the past, now it is changed. You will use hardware Serial of Python (DFL168A.Ser4DFL168A) to send/ receive data to/from DEV1 of DFL168A directly. In general, DEV1 connects GPS module.

Syntax

r = DFL168A.beginTransparentSerial()

Parameters

Nothing

Returns

r : bool, Ture denotes we set up DFL168A.Ser4DFL168A for accessing DEV1 instead of AT command interface successfully.

4.3.8 endTransparentSerial Function

endTransparentSerial()

Description

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

Syntax

r = DFL168A.endTransparentSerial()

Parameters

Nothing

Returns

r : bool, Ture denotes we resume Python Serial port to AT command interface successfully.

4.3.9 setExitTransparentKey Function

setExitTransparentKey(EndTransparentChar)

Description

setExitTransparentKey() will set up character for exiting Transparent Serial mode. When hardware Serial port of DFL168A, which was used by DFL168A interface command, and is used for accessing DEV1, receives this character from Python, DFL168A will resume the hardware Serial port to use for DFL168A interface Command. This function return true when setting succeed. This method can't execute before running begin method.

Syntax

r = DFL168A.setExitTransparentKey(EndTransparentChar)

Parameters

EndTransparentChar: the first parameter, int type, this is input parameter. It is a character ascii code.

Returns

r : bool, Ture denotes we set up exit character successfully.

4.3.10 setSleepDelay Function

setSleepDelay(SleepDelay_inSeconds)

Description

setSleepDelay(SleepDelay_inSeconds) 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 function once every 18.2 or less hours. This function will return true if call success, otherwise return false.

Syntax

r = DFL168A.setSleepDelay(SleepDelay_inSeconds)

Parameters

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

Returns

r : bool, Ture denotes we set up delay time successfully.

4.4 J1939 sub-module

4.4.1 Sub-sub-modules

J1939 has the following Sub-Modules:

PGN65267

PGN65262

PGN65256

PGN65269

PGN65257

PGN61444

PGN61443

PGN65270

PGN65271

PGN65272

PGN65266

PGN65263

PGN65253

PGN65214

PGN65248

PGN65276

PGN65265

PGN57344

PGN64996

PGN61445

PGN65268

We will explain these sub-modules later.

4.4.2 Function

getVIN()
getDTC(DTCFormat=1)
clearDTC()

All function return at least one value. The first return value is bool type, which tells us whether this function call successfully.
The other return values are truly results we want from calling this function.

4.4.2.1 getVIN function

getVIN()

Description

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

Syntax

r, VIN = DFL168A.J1939.getVIN()

Parameters

Nothing.

Returns
r : bool, Ture denotes we get VIN successfully.
VIN: string , It is vehicle identification number

4.4.2.2 getDTC function

getDTC(DTCFormat=1)

Description

This function will get DTC information of vehicle. It will return true if success, otherwise return false.

Syntax

r, DTC_Num, SPN, FMI, OC = DFL168A.J1939.getDTC(DTCFormat=1)

Parameters

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

Returns

r : bool, Ture denotes we get DTC successfully

DTC_Num: int type, it is quantity of vehicle DTC.

SPN: list type, list element is SPN number of vehicle.

FMI: list type, list element is FMI of vehicle.

OC: list type, list element is OC of vehicle.

4.4.2.3 clearDTC function

clearDTC()

Description

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

Syntax

r = DFL168A.J1939.clearDTC()

Parameters

Nothing

Returns

r : bool, Ture denotes we clear fault successfully

4.4.3 Sub-module PGN65267

4.4.3.1 Function

refresh()

getLatitude( )

getLongitude( )

4.4.3.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65267.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.3.1.2 getLatitude function

getLatitude()

Description

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

Syntax

r, Latitude = DFL168A.J1939.PGN65267.getLatitude( )

Parameters

.Nothing

Returns

r : bool, Ture denotes getting latitude successfully

Latitude: float type, It is latitude of vehicle location in degree unit

4.4.3.1.3 getLongitude function

getLongitude()

Description

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

Syntax

r, Longitude = DFL168A.J1939.PGN65267.getLongitude()

Parameters

Nothing

Returns

r : bool, Ture denotes getting longitude successfully

Longitude: float type, It is longitude of vehicle location in degree unit.

4.4.4 Sub-module PGN65262

4.4.4.1 Function

refresh()

getCoolantTemperature( )

getFuelTemp( )

getOilTemp( )

4.4.4.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65262.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.4.1.2 getCoolantTemperature function

getCoolantTemperature()

Description

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

Syntax

r, temp = DFL168A.J1939.PGN65262.getCoolantTemperature()

Parameters

Nothing.

Returns

r : bool, Ture denotes getting coolant temperature successfully

temp: int type, It is engine coolant temperature in Celsius degree.

4.4.4.1.3 getFuelTemp function

getFuelTemp( )

Description

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

Syntax

r, temp = DFL168A.J1939.PGN65262.getFuelTemp( )

Parameters

Nothing.

Returns

r : bool, Ture denotes getting fuel temperature successfully

temp: int type, It is fuel temperature in Celsius degree.

4.4.4.1.4 getOilTemp function

getOilTemp( )

Description

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

Syntax

r, temp = DFL168A.J1939.PGN65262.getOilTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting engine oil temperature successfully

temp: float type, It is engine oil temperature in Celsius degree.

4.4.5 Sub-module PGN65256

4.4.5.1 Function

refresh( )

getAltitude( )

getNavBasedSpeed( )

4.4.5.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65256.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.5.1.2 getAltitude function

getAltitude( )

Description

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

Syntax

r, Altitude = DFL168A.J1939.PGN65256.getAltitude( )

Parameters

Nothing.

Returns

r : bool, Ture denotes getting altitude successfully

Altitude: float type, It is altitude of vehicle location.

4.4.5.1.3 getNavBasedSpeed function

getNavBasedSpeed()

Description

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

Syntax

r, Speed = DFL168A.J1939.PGN65256.getNavBasedSpeed( )

Parameters

Nothing.

Returns

r : bool, Ture denotes getting vehicle speed successfully

Speed: float type, It is vehicle speed based on navigation in Km/h

4.4.6 Sub-module PGN65269

4.4.6.1 Function

refresh()

getBarometricPressure( )

getAmbientTemp( )

getInletTemp( )

getRoadTemp( )

getCabInteriorTemp( )

4.4.6.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65269.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.6.1.2 getBarometricPressure function

getBarometricPressure( )

Description

PGN65269.getBarometricPressure( ) will get Barometric Pressure in KPa. It will return true if

success, otherwise return false.

Syntax

r, BaroPressure = DFL168A.J1939.PGN65269.getBarometricPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting barometric pressure successfully

BaroPressure: float type, It is Barometric Pressure in kPa

4.4.6.1.3 getAmbientTemp function

getAmbientTemp( )

Description

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

Syntax

r, AmbientTemp = DFL168A.J1939.PGN65269.getAmbientTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ambient air temperature successfully

AmbientTemp: float type, It is Ambient Air Temperature in Celsius degree.

4.4.6.1.4 getInletTemp function

getInletTemp( )

Description

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

Syntax

r, InletTemp = DFL168A.J1939.PGN65269.getInletTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Air Inlet Temperature successfully

InletTemp: int type, It is air inlet temperature in Celsius degree.

4.4.6.1.5 getRoadTemp function

getRoadTemp( )

Description

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

Syntax

r, RoadTemp = DFL168A.J1939.PGN65269.getRoadTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting road temperature successfully

RoadTemp: float type, It is road temperature in Celsius degree.

4.4.6.1.6 getCabInteriorTemp function

getCabInteriorTemp( )

Description

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

Syntax

r, CabInteriorTemp = DFL168A.J1939.PGN65269.getCabInteriorTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting cab Interior temperature successfully

CabInteriorTemp: float type, It is Cab Interior Temperature in Celsius degree.

4.4.7 Sub-module PGN65257

4.4.7.1 Function

refresh()

getEngineTripFuel( )

getEngineTotalFuelUsed()

4.4.7.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65257.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.7.1.2 getEngineTripFuel function

getEngineTripFuel( )

Description

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

Syntax

r, EngineTripFuel = DFL168A.J1939.PGN65257.getEngineTripFuel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Trip Fuel successfully

EngineTripFuel: float type, It is Engine Trip Fuel in L.

4.4.7.1.3 getEngineTotalFuelUsed function

getEngineTotalFuelUsed( )

Description

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

Syntax

r, EngineTotalFuelUsed = DFL168A.J1939.PGN65257.getEngineTotalFuelUsed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Total Fuel Used successfully

EngineTotalFuelUsed: float type, It is Engine Total Fuel Used in L.

4.4.8 Sub-module PGN61444

4.4.8.1 Function

refresh( )

getActualEngineTorque( )

getEngineSpeed( )

4.4.8.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN61444.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.8.1.2 getActualEngineTorque function

getActualEngineTorque( )

Description

PGN61444.getActualEngineTorque( ) will get Actual Engine - Percent Torque. It will return true if success, otherwise return false.

Syntax

r, ActualEngineTorque = DFL168A.J1939.PGN61444.getActualEngineTorque( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Actual Engine - Percent Torque successfully

ActualEngineTorque: int type, It is Actual Engine - Percent Torque.

4.4.8.1.3 getEngineSpeed function

getEngineSpeed()

Description

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

Syntax

r, EngineSpeed = DFL168A.J1939.PGN61444.getEngineSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting engine speed successfully

EngineSpeed: float type, It is engine speed in rpm.

4.4.9 Sub-module PGN61443

4.4.9.1 Function

refresh()

getAccelPedalPosi1( )

getAccelPedalPosi2( )

getEnginePerLoadAtCurrSpeed()

4.4.9.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN61443.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.9.1.2 getAccelPedalPosi1 function

getAccelPedalPosi1( )

Description

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

Syntax

r, AccelPedalPosi1 = DFL168A.J1939.PGN61443.getAccelPedalPosi1( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Accelerator Pedal Position 1 successfully

AccelPedalPosi1: float type, It is Accelerator Pedal Position 1 in percentage.

4.4.9.1.3 getAccelPedalPosi2 function

getAccelPedalPosi2( )

Description

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

Syntax

r, AccelPedalPosi2 = DFL168A.J1939.PGN61443.getAccelPedalPosi2( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Accelerator Pedal Position 2 successfully

AccelPedalPosi2: float type, It is Accelerator Pedal Position 2 in percentage.

4.4.9.1.4 getEnginePerLoadAtCurrSpeed function

getEnginePerLoadAtCurrSpeed( )

Description

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

Syntax

r, EnginePerLoadAtCurrSpeed = DFL168A.J1939.PGN61443.getEnginePerLoadAtCurrSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Percent Load At Current Speed successfully

EnginePerLoadAtCurrSpeed: int type, It is Engine Percent Load At Current Speed.

4.4.10 Sub-module PGN65270

4.4.10.1 Function

refresh( )

getIntakeManifoldPressure( )

getIntakeManifoldTemp( )

getEngineAirInletPressure( )

getEngineExhaustGasTemp( )

getEngineAirFilterDiffPressure( )

4.4.10.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65270.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.10.1.2 getIntakeManifoldPressure function

getIntakeManifoldPressure( )

Description

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

Syntax

r, IntakeManifoldPressure =DFL168A.J1939.PGN65270.getIntakeManifoldPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Intake Manifold #1 Pressure successfully

IntakeManifoldPressure: int type, It is Engine Intake Manifold #1 Pressure in kPa.

4.4.10.1.3 getIntakeManifoldTemp function

getIntakeManifoldTemp( )

Description

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

Syntax

r, IntakeManifoldTemp = DFL168A.J1939.PGN65270.getIntakeManifoldTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Intake Manifold 1 Temperature successfully

IntakeManifoldTemp: int type, It is Engine Intake Manifold 1 Temperature in Celsius degree.

4.4.10.1.4 getEngineAirInletPressure function

getEngineAirInletPressure()

Description

PGN65270.getEngineAirInletPressure( ) will get Engine Air Inlet Pressure in kPa. It will return true if

success, otherwise return false.

Syntax

r, EngineAirInletPressure = DFL168A.J1939.PGN65270.getEngineAirInletPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Air Inlet Pressure successfully

EngineAirInletPressure: int type, It is Engine Air Inlet Pressure in kPa.

4.4.10.1.5 getEngineExhaustGasTemp function

getEngineExhaustGasTemp( )

Description

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

Syntax

r, EngineExhaustGasTemp = DFL168A.J1939.PGN65270.getEngineExhaustGasTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Exhaust Gas Temperature successfully

EngineExhaustGasTemp: float type, It is Engine Exhaust Gas Temperature in Celsius degree.

4.4.10.1.6 getEngineAirFilterDiffPressure function

getEngineAirFilterDiffPressure( )

Description

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

Syntax

r, EngineAirFilterDiffPressure = DFL168A.J1939.PGN65270.getEngineAirFilterDiffPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Air Filter 1 Differential Pressure successfully

EngineAirFilterDiffPressure: float type, It is Engine Air Filter 1 Differential Pressure in kPa.

4.4.11 Sub-module PGN65271

4.4.11.1 Function

refresh()

getAlternatorVoltage( )

getElectricalVoltage( )

getBatteryVoltage( )

4.4.11.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65271.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.11.1.2 getAlternatorVoltage function

getAlternatorVoltage( )

Description

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

Syntax

r, AlternatorVoltage = DFL168A.J1939.PGN65271.getAlternatorVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Charging System Potential (Voltage) successfully

AlternatorVoltage: float type, It is Charging System Potential (Voltage).

4.4.11.1.3 getElectricalVoltage function

getElectricalVoltage( )

Description

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

Syntax

r, ElectricalVoltage = DFL168A.J1939.PGN65271.getElectricalVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Battery Potential / Power Input 1 successfully

ElectricalVoltage: float type, It is Battery Potential / Power Input 1.

4.4.11.1.4 getBatteryVoltage function

getBatteryVoltage( )

Description

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

Syntax

r, BatteryVoltage = DFL168A.J1939.PGN65271.getBatteryVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Keyswitch Battery Potential successfully

BatteryVoltage: float type, It is Keyswitch Battery Potential.

4.4.12 Sub-module PGN65272

4.4.12.1 Function

refresh()

getTransmissionOilLevel( )

getTransmissionOilLevelHighLow( )

getTransmissionOilPressure( )

getTransmissionOilTemp( )

4.4.12.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65272.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.12.1.2 getTransmissionOilLevel function

getTransmissionOilLevel( )

Description

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

Syntax

r, Percent = DFL168A.J1939.PGN65272.getTransmissionOilLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Transmission Oil Level successfully

Percent: float type, It is Transmission Oil Level in percentage.

4.4.12.1.3 getTransmissionOilLevelHighLow function

getTransmissionOilLevelHighLow( )

Description

PGN65272.getTransmissionOilLevelHighLow( ) 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

r, HighLow = DFL168A.J1939.PGN65272.getTransmissionOilLevelHighLow( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Amount of current volume of transmission sump oil successfully

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

4.4.12.1.4 getTransmissionOilPressure function

getTransmissionOilPressure( )

Description

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

Syntax

r, Pressure = DFL168A.J1939.PGN65272.getTransmissionOilPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Transmission Oil Pressure successfully

Pressure: float type, It is Transmission Oil Pressure in kPa.

4.4.12.1.5 getTransmissionOilTemp function

getTransmissionOilTemp( )

Description

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

Syntax

r, Temperature = DFL168A.J1939.PGN65272.getTransmissionOilTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Transmission Oil Temperature successfully

Temperature: float type, It is Transmission Oil Temperature in Celsius degree.

4.4.13 Sub-module PGN65266

4.4.13.1 Function

refresh();

getFuelRate( )

getInstantFuelEconomy( )

getAvgFuelEconomy( )

getEngineThrottlePos( )

4.4.13.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65266.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.13.1.2 getFuelRate function

getFuelRate( )

Description

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

return false.

Syntax

r, FuelRate = DFL168A.J1939.PGN65266.getFuelRate( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Fuel Rate successfully

FuelRate: float type, It is Engine Fuel Rate in L/H.

4.4.13.1.3 getInstantFuelEconomy function

getInstantFuelEconomy( )

Description

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

Syntax

r, InstantFuelEconomy = DFL168A.J1939.PGN65266.getInstantFuelEconomy( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Instantaneous Fuel Economy successfully

InstantFuelEconomy: float type, It is Engine Instantaneous Fuel Economy in Km/L.

4.4.13.1.4 getAvgFuelEconomy function

getAvgFuelEconomy( )

Description

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

Syntax

r, AvgFuelEconomy = DFL168A.J1939.PGN65266.getAvgFuelEconomy( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Average Fuel Economy successfully
AvgFuelEconomy: float type, It is Engine Average Fuel Economy in Km/L.

4.4.13.1.5 getEngineThrottlePos function

getEngineThrottlePos( )

Description

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

Syntax

r, EngineThrottlePos = DFL168A.J1939.PGN65266.getEngineThrottlePos( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Throttle Position successfully
EngineThrottlePos: float type, It is Engine Throttle Position in percentage.

4.4.14 Sub-module PGN65263

4.4.14.1 Function

refresh()
getFueDeliveryPressure( )
getEngineOilLevel( )
getEngineOilPressure( )
getEngineCoolantPressure( )
getEngineCoolantLevel( )

4.4.14.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65263.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.14.1.2 getFueDeliveryPressure function

getFueDeliveryPressure( )

Description

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

Syntax

r, FueDeliveryPressure = DFL168A.J1939.PGN65263.getFueDeliveryPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Fuel Delivery Pressure successfully

FueDeliveryPressure: int type, It is Engine Fuel Delivery Pressure in kPa.

4.4.14.1.3 getEngineOilLevel function

```
getEngineOilLevel()
```

Description

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

Syntax

r, EngineOilLevel = DFL168A.J1939.PGN65263.getEngineOilLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Oil Level successfully

EngineOilLevel: float type, It is Engine Oil Level in percentage

4.4.14.1.4 getEngineOilPressure function

getEngineOilPressure( )

Description

PGN65263.getEngineOilPressure( ) will get Engine Oil Pressure in kPa. It will return true if success,

otherwise return false.

Syntax

r, EngineOilPressure = DFL168A.J1939.PGN65263.getEngineOilPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Oil Pressure successfully

EngineOilPressure: int type, It is Engine Oil Pressure in kPa.

4.4.14.1.5 getEngineCoolantPressure function

getEngineCoolantPressure( )

Description

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

Syntax

r, EngineCoolantPressure = DFL168A.J1939.PGN65263.getEngineCoolantPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Coolant Pressure successfully

EngineCoolantPressure: int type, It is Engine Coolant Pressure in kPa.

4.4.14.1.6 getEngineCoolantLevel function

getEngineCoolantLevel( )

Description

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

Syntax

r, EngineCoolantLevel = DFL168A.J1939.PGN65263.getEngineCoolantLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Coolant Level successfully

EngineCoolantLevel: float type, It is Engine Coolant Level in percentage

4.4.15 Sub-module PGN65253

4.4.15.1 Function

refresh( )

getTotalEngineHours( )

getTotalEngineRevolutions( )

4.4.15.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65253.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.15.1.2 getTotalEngineHours function

getTotalEngineHours( )

Description

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

Syntax

r, TotalEngineHours = DFL168A.J1939.PGN65253.getTotalEngineHours( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Total Hours of Operation successfully

TotalEngineHours: float type, It is Engine Total Hours of Operation.

4.4.15.1.3 getTotalEngineRevolutions function

getTotalEngineRevolutions( )

Description

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

Syntax

r, TotalEngineRevolutions = DFL168A.J1939.PGN65253.getTotalEngineRevolutions( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Total Revolutions successfully

TotalEngineRevolutions: float type, It is Engine Total Revolutions. Unit is r.

4.4.16 Sub-module PGN65214

4.4.16.1 Function

refresh()

getRatedEngineSpeed( )

4.4.16.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65214.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.16.1.2 getRatedEngineSpeed function

getRatedEngineSpeed( )

Description

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

Syntax

r, RatedEngineSpeed = DFL168A.J1939.PGN65214.getRatedEngineSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Engine Rated Speed successfully

RatedEngineSpeed: float type, It is Engine Rated Speed in rpm.

4.4.17 Sub-module PGN65248

4.4.17.1 Function

refresh()

getTripDistance( )

getTotalDistance( )

4.4.17.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65248.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.17.1.2 getTripDistance function

getTripDistance( )

Description

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

return false.

Syntax

r, TripDistance = DFL168A.J1939.PGN65248.getTripDistance( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Trip Distance successfully

TripDistance: float type, It is Trip Distance in Km.

4.4.17.1.3 getTotalDistance function

getTotalDistance( )

Description

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

Syntax

r, TotalDistance = DFL168A.J1939.PGN65248.getTotalDistance( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Total Vehicle Distance successfully

TotalDistance: float type, It is Total Vehicle Distance in Km.

4.4.18 Sub-module PGN65276

4.4.18.1 Function

refresh()

getWasherFluidLevel( )

getFuelLevel1( )

getFuelLevel2( )

4.4.18.1.1 refresh function

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 functions in this PGN module, you

should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65276.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.18.1.2 getWasherFluidLevel function

getWasherFluidLevel( )

Description

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

Syntax

r, WasherFluidLevel = DFL168A.J1939.PGN65276.getWasherFluidLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Washer Fluid Level successfully

WasherFluidLevel: float type, It is Washer Fluid Level in percentage

4.4.18.1.3 getFuelLevel1 function

getFuelLevel1( )

Description

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

Syntax

r, FuelLevel1 = DFL168A.J1939.PGN65276.getFuelLevel1( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Fuel Level 1 successfully

FuelLevel1: float type, It is Fuel Level 1 in percentage

4.4.18.1.4 getFuelLevel2 function

getFuelLevel2( )

Description

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

Syntax

r, FuelLevel2 = DFL168A.J1939.PGN65276.getFuelLevel2( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Fuel Level 2 successfully

FuelLevel2: float type, It is Fuel Level 2 in percentage

4.4.19 Sub-module PGN65265

4.4.19.1 Function

refresh()

getWheelBasedVehicleSpeed( )

getParkingBrake( )

getBrake( )

4.4.19.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN65265.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.19.1.2 getWheelBasedVehicleSpeed function

getWheelBasedVehicleSpeed( )

Description

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

Syntax

r, WheelBasedVehicleSpeed = DFL168A.J1939.PGN65265.getWheelBasedVehicleSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Wheel-Based Vehicle Speed successfully

WheelBasedVehicleSpeed: float type, It is Wheel-Based Vehicle Speed in km/h.

4.4.19.1.3 getParkingBrake function

getParkingBrake( )

Description

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

Syntax

r, ParkingBrakeSet = DFL168A.J1939.PGN65265.getParkingBrake( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Parking brake switch status successfully

ParkingBrakeSet: bool type, It denotes whether Parking brake switch set.

4.4.19.1.4 getBrake function

getBrake( )

Description

PGN65265.getBrake( ) 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

r, BrakePedalDepressed = DFL168A.J1939.getBrake( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Brake switch status successfully

BrakePedalDepressed: bool type, It denotes whether Brake pedal depressed.

4.4.20 Sub-module PGN57344

4.4.20.1 Function

refresh()

getSeatBelt( )

4.4.20.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN57344.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.20.1.2 getSeatBelt function

getSeatBelt( )

Description

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

Syntax

r, buckled = DFL168A.J1939.PGN57344.getSeatBelt( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting status of Seat Belt Switch successfully

buckled: bool type, true means that Seat Belt is buckled, false means that Seat Belt is not buckled

4.4.21 Sub-module PGN64996

4.4.21.1 Function

refresh()

getPayLoad( )

4.4.21.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN64996.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.21.1.2 getSeatBelt function

getPayLoad( )

Description

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

Syntax

r, PayLoad = DFL168A.J1939.PGN64996.getPayLoad( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Payload Percentage successfully

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

4.4.22 Sub-module PGN61445

4.4.22.1 Function

refresh()

getCurrentGear( )

getSelectedGear( )

4.4.22.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN61445.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.22.1.2 getCurrentGear function

getCurrentGear( )

Description

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

Syntax

r, CurrentGear = DFL168A.J1939.PGN61445.getCurrentGear( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Transmission Current Gear successfully

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

4.4.22.1.3 getSelectedGear function

getSelectedGear( )

Description

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

Syntax

r, SelectedGear = DFL168A.J1939.PGN61445.getSelectedGear( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Transmission Selected Gear successfully

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

4.4.23 Sub-module PGN65268

4.4.23.1 Function

refresh()

getTirePressure( )

getTireTemperature( )

getTireLocation( )

getTireValvePressureMonitor( )

4.4.23.1.1 refresh function

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 functions in this PGN module, you should call this function firstly.

Syntax

r = DFL168A.J1939.PGN61445.refresh()

Parameters

Nothing

Returns

r : bool, Ture denotes PGN data update successfully

4.4.23.1.2 getTirePressure function

getTirePressure( )

Description

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

Syntax

r, TirePressure = DFL168A.J1939.PGN65268.getTirePressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting tire pressure successfully

TirePressure: int type, Data rang is 0 to 1000 KPa.

4.4.23.1.3 getTireTemperature function

getTireTemperature( )

Description

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

Syntax

r, Temperature = DFL168A.J1939.PGN65268.getTireTemperature( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting tire temperature successfully

Temperature: float type, Data range is -273 to 1734.96875 deg C

4.4.23.1.4 getTireLocation function

getTireLocation( )

Description

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

Syntax

r, Front2RearNumber, Left2RighNumber = DFL168A.J1939.PGN65268.getTireLocation( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting tire location successfully

Front2RearNumber: int type, it represents a position number , counting front to rear on the vehicle.

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

4.4.23.1.5 getTireValvePressureMonitor function

getTireValvePressureMonitor( )

Description

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

Syntax

r, TireValvePressureMonitor = DFL168A.J1939.PGN65268.getTireValvePressureMonitor( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the pressure level of tire successfully

TireValvePressureMonitor: int type, 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.

4.5 Sub-module J1708

4.5.1 Function

getAirPressure( )

getEngineOilPressure( )

getEngineCoolantPressure( )
getFuelLevel1( )
getFuelLevel2( )
getBarometricPressure( )
getEngineThrottlePos( )
getWasherFluidLevel( )
getVehicleSpeed( )
getAccelPedalPosi1( )
getAccelPedalPosi2( )
getAccelPedalPosi3( )
getEngineLoad( )
getEngineOilLevel( )
getCoolantTemperature( )
getEngineCoolantLevel( )
getTransmissionOilLevel( )
getTransmissionOilLevelHighLow( )
getTransmissionOilPressure( )
getTransmissionOilTemp( )
getPowerSpecificInstantFuelEconomy( )
getAvgFuelRate( )
getInstantFuelEconomy( )
getAvgFuelEconomy( )
getElectricalVoltage( )
getRatedEnginePower( )
getBatteryVoltage( )
getAlternatorVoltage( )
getAmbientTemp( )
getCargoAmbientTemp( )
getRoadTemp( )
getCabInteriorTemp( )
getInletTemp( )
getFuelTemp( )
getOilTemp( )
getCargoWeight( )
getEngineTripFuel( )
getEngineTotalFuelUsed( )
getFuelRate( )
getRatedEngineSpeed( )
getEngineSpeed( )

getIntakeManifoldTemp( )
getPowerTakeoffStatus( )
getTripDistance( )
getTotalDistance( )
getTotalEngineHours( )
getTotalEngineRevolutions( )
getVIN( )
getDTC( )
clearDTC(MID,PID_SID, IsPID);
getFaultDescription(MID,PID_SID, IsPID, FMI)
getPIDSIDDescription MID, PID_SID, IsPID)

All function return at least one value. The first return value is bool type, which tells us whether this function call successfully.
The other return values are truly results we want from calling this function.

4.5.1.1 getAirPressure Function

getAirPressure( )

Description

getAirPressure( ) 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

r, AirPressure = DFL168A.J1708.getAirPressure( )

Parameters

Nothing.

Returns
r : bool, Ture denotes getting Gauge Pressure of air in system successfully
AirPressure:, float type, It is Gauge Pressure of air in system in kPa.

4.5.1.2 getEngineOilPressure Function

getEngineOilPressure( )

Description

getEngineOilPressure( ) 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

r, EngineOilPressure = DFL168A.J1708.getEngineOilPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Gauge pressure of oil in the engine lubrication system successfully

EngineOilPressure: float type, It is Gauge pressure of oil in the engine lubrication system in kPa.

4.5.1.3 getEngineCoolantPressure Function

getEngineCoolantPressure( )

Description

getEngineCoolantPressure( ) 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

r, EngineCoolantPressure = DFL168A.J1708.getEngineCoolantPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Gauge pressure of liquid found in the engine cooling system successfully

EngineCoolantPressure: float type, It is Gauge pressure of liquid found in the engine cooling system in kPa.

4.5.1.4 getFuelLevel1 Function

getFuelLevel1( )

Description

getFuelLevel1( ) 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

r, FuelLevel1 = DFL168A.J1708.getFuelLevel1( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of volume of fuel to the total volume of the primary fuel storage container successfully

FuelLevel1: float type, It is ratio of volume of fuel to the total volume of the primary fuel storage

container. Unit is percentage.

4.5.1.5 getFuelLevel2 Function

bool getFuelLevel2( )

Description

getFuelLevel2( ) 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

r, FuelLevel2 = DFL168A.J1708.getFuelLevel2( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of volume of fuel to the total volume of the second fuel storage container successfully

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

4.5.1.6 getBarometricPressure Function

getBarometricPressure( )

Description

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

Syntax

r, Pressure = DFL168A.J1708.getBarometricPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting absolute air pressure of the atmosphere successfully

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

4.5.1.7 getEngineThrottlePos Function

getEngineThrottlePos( )

Description

getEngineThrottlePos( ) 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

r, EngineThrottlePos = DFL168A.J1708.getEngineThrottlePos( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the position of the valve used to regulate the supply of a fluid successfully

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

4.5.1.8 getWasherFluidLevel Function

getWasherFluidLevel( )

Description

getWasherFluidLevel( ) 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

r, WasherFluidLevel = DFL168A.J1708.getWasherFluidLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of volume of liquid to total container volume of fluid reservoir in windshield wash system successfully

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

4.5.1.9 getVehicleSpeed Function

getVehicleSpeed( )

Description

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

Syntax

r, VehicleSpeed = DFL168A.J1708.getVehicleSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting vehicle road speed successfully

WasherFluidLevel: float type, It is vehicle road speed in km/h.

4.5.1.10 getAccelPedalPosi1 Function

getAccelPedalPosi1( )

Description

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

Syntax

r, AccelPedalPosi1 = DFL168A.J1708.getAccelPedalPosi1( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of actual accelerator pedal position to maximum pedal position successfully

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

4.5.1.11 getAccelPedalPosi2 Function

getAccelPedalPosi2( )

Description

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

Syntax

r, AccelPedalPosi2 = DFL168A.J1708.getAccelPedalPosi2( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of actual accelerator pedal position to maximum pedal position successfully

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

4.5.1.12 getAccelPedalPosi3 Function

getAccelPedalPosi3( )

Description

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

Syntax

r, AccelPedalPosi3 = DFL168A.J1708.getAccelPedalPosi3( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of actual accelerator pedal position to maximum pedal position successfully

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

4.5.1.13 getEngineLoad Function

getEngineLoad( )

Description

getEngineLoad( ) 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

r, Percent = DFL168A.J1708.getEngineLoad( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of current output torque to maximum torque available at the current engine speed successfully

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

4.5.1.14 getEngineOilLevel Function

getEngineOilLevel( )

Description

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

Syntax

r, EngineOilLevel = DFL168A.J1708.getEngineOilLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of current volume of engine sump oil to maximum required volume successfully
EngineOilLevel: float type, It is the ratio of current volume of engine sump oil to maximum required volume. This is percentage.

4.5.1.15 getCoolantTemperature Function

getCoolantTemperature( )

Description

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

Syntax

r, temp = DFL168A.J1708.getCoolantTemperature( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the temperature of liquid found in engine cooling system successfully
temp: float type, It is the temperature of liquid found in engine cooling system in Celsius degree.

4.5.1.16 getEngineCoolantLevel Function

getEngineCoolantLevel( )

Description

getEngineCoolantLevel( ) 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

r, EngineCoolantLevel = DFL168A.J1708.getEngineCoolantLevel( )

Parameters

Nothing.

Returns

r : bool, Ture denotes getting ratio of volume of liquid found in engine cooling system to total

cooling system volume successfully

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

4.5.1.17 getTransmissionOilLevel Function

getTransmissionOilLevel( )

Description

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

Syntax

r, Percent = DFL168A.J1708.getTransmissionOilLevel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting ratio of volume of transmission sump oil to recommended volume successfully

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

4.5.1.18 getTransmissionOilLevelHighLow Function

getTransmissionOilLevelHighLow( )

Description

getTransmissionOilLevelHighLow( ) 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

r, HighLow = DFL168A.J1708.getTransmissionOilLevelHighLow( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting amount of current volume of transmission sump oil compared to recommended volume successfully

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

4.5.1.19 getTransmissionOilPressure Function

getTransmissionOilPressure( )

Description

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

Syntax

r, Pressure = DFL168A.J1708.getTransmissionOilPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting gage pressure of lubrication fluid in transmission successfully

Pressure: float type, It is gage pressure of lubrication fluid in transmission, measured after pump in kPa.

4.5.1.20 getTransmissionOilTemp Function

getTransmissionOilTemp( )

Description

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

Syntax

r, Temperature = DFL168A.J1708.getTransmissionOilTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of transmission lubricant successfully

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

4.5.1.21 getPowerSpecificInstantFuelEconomy Function

getPowerSpecificInstantFuelEconomy( )

Description

getPowerSpecificInstantFuelEconomy( ) 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

r, Rate = DFL168A.J1708.getPowerSpecificInstantFuelEconomy( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting instantaneous fuel economy of the engine successfully

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

4.5.1.22 getAvgFuelRate Function

getAvgFuelRate( )

Description

getAvgFuelRate( ) 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

r, FuelRate = DFL168A.J1708.getAvgFuelRate( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting continuous averaging fuel per hour per segment of engine operation in L/s successfully

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

4.5.1.23 getInstantFuelEconomy Function

getInstantFuelEconomy( )

Description

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

Syntax

r, InstantFuelEconomy = DFL168A.J1708.getInstantFuelEconomy( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting current fuel economy at current vehicle velocity successfully

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

4.5.1.24 getAvgFuelEconomy Function

getAvgFuelEconomy( )

Description

getAvgFuelEconomy( ) 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

r, AvgFuelEconomy = DFL168A.J1708.getAvgFuelEconomy( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting average of instantaneous fuel economy for that segment of vehicle operation of interest successfully

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

4.5.1.25 getElectricalVoltage Function

getElectricalVoltage( )

Description

getElectricalVoltage( ) 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

r, BatteryVoltage = DFL168A.J1708.getElectricalVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting electrical potential measured at the input of the electronic control unit supplied through a switching device successfully

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

4.5.1.26 getRatedEnginePower Function

getRatedEnginePower( )

Description

getRatedEnginePower( ) 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

r, Power = DFL168A.J1708.getRatedEnginePower( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting net brake power that the engine will deliver continuously, specified for a given application at a rated speed successfully

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

4.5.1.27 getBatteryVoltage Function

getBatteryVoltage( )

Description

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

Syntax

r, BatteryVoltage = DFL168A.J1708.getBatteryVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting measured electrical potential of the battery successfully

BatteryVoltage: float type, It is the measured electrical potential of the battery.

4.5.1.28 getAlternatorVoltage Function

getAlternatorVoltage( )

Description

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

Syntax

r, AlternatorVoltage = DFL168A.J1708.getAlternatorVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting measured electrical potential of the alternator successfully

AlternatorVoltage: float type, It is the measured electrical potential of the alternator.

4.5.1.29 getAmbientTemp Function

getAmbientTemp( )

Description

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

Syntax

r, AmbientTemp = DFL168A.J1708.getAmbientTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of air surrounding vehicle successfully

AmbientTemp: float type, It is the temperature of air surrounding vehicle in Celsius degree.

4.5.1.30 getCargoAmbientTemp Function

getCargoAmbientTemp( )

Description

getCargoAmbientTemp( ) 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

r, CargoTemp = DFL168A.J1708. getCargoAmbientTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of air inside vehicle container successfully

CargoTemp: float type, It is the temperature of air inside vehicle container used to accommodate cargo in Celsius degree.

4.5.1.31 getRoadTemp Function

getRoadTemp( )

Description

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

Syntax

r, RoadTemp = DFL168A.J1708.getRoadTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of road surface, over which vehicle is operating, successfully

RoadTemp: float type, It is the temperature of road surface over which vehicle is operating in Celsius degree.

4.5.1.32 getCabInteriorTemp Function

getCabInteriorTemp( )

Description

getCabInteriorTemp( ) 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

r, CabInteriorTemp = DFL168A.J1708.getCabInteriorTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of air inside the part of the vehicle successfully

CabInteriorTemp: float type, It is the temperature of air inside the part of the vehicle in Celsius degree.

4.5.1.33 getInletTemp Function

getInletTemp( )

Description

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

Syntax

r, InletTemp = DFL168A.J1708.getInletTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of air entering vehicle air induction system successfully

InletTemp: float type, It is the temperature of air entering vehicle air induction system in Celsius degree.

4.5.1.34 getFuelTemp Function

getFuelTemp( )

Description

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

Syntax

r, temp = DFL168A.J1708.getFuelTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of fuel entering injectors successfully

temp: float type, It is the temperature of fuel entering injectors in Celsius degree.

4.5.1.35 getOilTemp Function

getOilTemp( )

Description

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

Syntax

r, temp = DFL168A.J1708.getOilTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting temperature of engine lubricant in system successfully

temp: float type, It is the temperature of engine lubricant in Celsius degree.

4.5.1.36 getCargoWeight Function

getCargoWeight( )

Description

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

otherwise return false.

Syntax

r, CargoW = DFL168A.J1708.getCargoWeight( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the force of gravity of freight carried successfully

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

4.5.1.37 getEngineTripFuel Function

getEngineTripFuel( )

Description

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

Syntax

r, EngineTripFuel = DFL168A.J1708.getEngineTripFuel( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the fuel consumed during all or part of a journey successfully

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

4.5.1.38 getEngineTotalFuelUsed Function

getEngineTotalFuelUsed( )

Description

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

Syntax

r, EngineTotalFuelUsed = DFL168A.J1708.getEngineTotalFuelUsed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the accumulated amount of fuel used during vehicle operation successfully

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

4.5.1.39 getFuelRate Function

getFuelRate( )

Description

getFuelRate( ) 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

r, FuelRate = DFL168A.J1708.getFuelRate( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the amount of fuel consumed by engine per unit of time successfully

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

4.5.1.40 getRatedEngineSpeed Function

getRatedEngineSpeed( )

Description

getRatedEngineSpeed( ) 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

r, RatedEngineSpeed = DFL168A.J1708.getRatedEngineSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the maximum governed rotational velocity of the engine crankshaft successfully

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

4.5.1.41 getEngineSpeed Function

getEngineSpeed( )

Description

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

Syntax

r, EngineSpeed = DFL168A.J1708.getEngineSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the rotational velocity of crankshaft successfully

EngineSpeed: float type, It is the rotational velocity of crankshaft in rpm.

4.5.1.42 getIntakeManifoldTemp Function

getIntakeManifoldTemp( )

Description

getIntakeManifoldTemp( ) 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

r, IntakeManifoldTemp = DFL168A.J1708.getIntakeManifoldTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the temperature of precombustion air found in intake manifold of engine air supply system successfully

IntakeManifoldTemp: float type, It is the temperature of precombustion air found in intake manifold of engine air supply system in Celsius degree.

4.5.1.43 getPowerTakeoffStatus Function

getPowerTakeoffStatus( )

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

r, PTOModeActive, ClutchSwitchOn, BrakeSwitchOn, AccelSwitchOn, ResumeSwitchOn, CoastSwitchOn, SetSwitchOn, PTOControlSwitchOn = DFL168A.

J1708.getPowerTakeoffStatus( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the state of the system used to transmit engine power to auxiliary equipment successfully

PTOModeActive: bool type, It is PTO mode. True means "active", False means "not active"

ClutchSwitchOn:bool type, It is clutch switch. True means "On", False means "Off"

BrakeSwitchOn: bool type, It is brake switch. True means "On", False means "Off"

AccelSwitchOn: bool type, It is accel switch. True means "On", False means "Off"

ResumeSwitchOn: bool type, It is resume switch. True means "On", False means "Off"

CoastSwitchOn: bool type, It is coast switch. True means "On", False means "Off"

SetSwitchOn: bool type, It is set switch. True means "On", False means "Off"

PTOControlSwitchOn: bool type, It is PTO control switch. True means "On", False means "Off"

4.5.1.44 getTripDistance Function

getTripDistance( )

Description

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

Syntax

r, TripDistance = DFL168A.J1708.getTripDistance( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the distance traveled during all or part of a journey successfully

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

4.5.1.45 getTotalDistance Function

getTotalDistance( )

Description

getTotalDistance( ) will get the accumulated distance travelled by vehicle during its operation in Km.

It will return true if success, otherwise return false.

Syntax

r, TotalDistance = DFL168A.J1708.getTotalDistance( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the accumulated distance travelled by vehicle successfully

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

4.5.1.46 getTotalEngineHours Function

getTotalEngineHours( )

Description

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

Syntax

r, TotalEngineHours = DFL168A.J1708.getTotalEngineHours( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the accumulated time of operation of engine successfully

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

4.5.1.47 getTotalEngineRevolutions Function

getTotalEngineRevolutions( )

Description

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

Syntax

r, TotalEngineRevolutions = DFL168A.J1708.getTotalEngineRevolutions( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the accumulated number of revolutions of engine crankshaft successfully

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

4.5.1.48 getVIN Function

getVIN( )

Description

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

Syntax

r, VIN = DFL168A.J1708.getVIN( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the Vehicle Identification Number successfully

VIN: String type, It is the Vehicle Identification Number.

4.5.1.49 getDTC Function

getDTC()

Description

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

Syntax

r, DTC_Num, MID, PID_SID, IsPID, FMI, IsActive, OccurrenceExist, OccurrenceCount = DFL168A.J1708.getDTC( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the diagnostic code and occurrence count successfully

DTC_Num: int type, It is the quantity of DTC.

MID: int type, It is MID of DTC.

PID_SID: list type, Element is int type, It is SID or PID of a standard diagnostic code. The next

returned value will decide it is SID or PID.

IsPID: list type, Element is bool type, It will tell us whether above returned value PID_SID is PID. True means "PID", False means "SID"

FMI: list type, Element is int type, It will tell us Failure mode identifier (FMI) of a standard diagnostic code.

IsActive: list type, Element is bool type, It will tell us whether fault is active. True means "active", False means "inactive"

OccurrenceExist: list type, Element is bool type, It will tell whether the next returned value OccurrenceCount exists. True means "exist", False means "not exist"

OccurrenceCount: list type, Element is int type, It will tell occurrence count for this diagnostic code

4.5.1.50 clearDTC Function

clearDTC(MID, PID_SID, IsPID)

Description

clearDTC(MID, PID_SID, 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

r = DFL168A.J1708.clearDTC(MID, PID_SID, IsPID)

Parameters

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

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

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

Returns

r : bool, Ture denotes clearing diagnostic codes successfully

4.5.1.51 getFaultDescription Function

getFaultDescription(MID, PID_SID, IsPID, FMI)

Description

getFaultDescription(MID, PID_SID, IsPID, FMI) 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

r, FaultDescription = DFL168A.J1708.getFaultDescription(MID, PID_SID, IsPID, FMI)

Parameters

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

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

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

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

Returns

r : bool, Ture denotes getting description string successfully

FaultDescription: String type, It is the string of described .DTC.

4.5.1.52 getPIDSIDDescription Function

getPIDSIDDescription(MID, PID_SID, IsPID)

Description

getPIDSIDDescription(MID, PID_SID, IsPID) 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

r, PID_SID_Description = DFL168A.J1708.getPIDSIDDescription(MID, PID_SID, IsPID)

Parameters

MID: the 1st parameter,byte type, It is the MID of device which some special PID or SID will be described..

PID_SID: the 2nd parameter,int type, It is the PID or SID. The next parameter will decide PID or SID.

IsPID: the 3rd parameter,bool type, It is used to identify the above parameter PID_SID. True means "PID", False means "SID"

Returns

r : bool, Ture denotes getting PID or SID description string successfully

PID_SID_Description: String type, It is the string of described PID or SID.

4.6 Sub-module ISO15765

4.6.1 Function

getCoolantTemperature( )
getEngineSpeed( )
getVehicleSpeed( )
getIntakeManifoldPressure( )
getFuelSystemStatus( )
getCalculatedLoadValue( )
getShortTermFuelTrimBank13( )
getLongTermFuelTrimBank13( )
getShortTermFuelTrimBank24( )
getLongTermFuelTrimBank24( )
getIgnitionTimingAdvance( )
getIntakeAirTemperature( )
getAirFlowRateFrmMAF( )
getAbsThrottlePosition( )
getOxygenSensorLocation( )
getBank1OSensor1Voltage( )
getBank1OSensor2Voltage( )
getBank1OSensor3Voltage( )
getBank1OSensor4Voltage( )
getBank2OSensor1Voltage( )
getBank2OSensor2Voltage( )
getBank2OSensor3Voltage( )
getBank2OSensor4Voltage( )
getBank3OSensor1Voltage( )
getBank3OSensor2Voltage( )
getBank4OSensor1Voltage( )
getBank4OSensor2Voltage( )
getOBDCertified( )
getTimeSinceEngineStart( )
getDistanceTraveledMIL( )
getFuelRailPressure( )
getFuelLevelInput( )
getDistanceTraveledSinceDTC_Clear( )
getBarometricPressure( )
getControlModuleVoltage( )
getRelativeThrottlePosition( )
getAmbientTemp( )

getCommandedThrottleActuatorControl( )
getEngineRunTimeMIL( )
getEngineRunTimeSinceDTC_Clear( )
getTypeOfFuelUsedCurrently( )
getRelativeAcceleratorPedalPosition( )
getHybridBatteryPackRemainingLife( )
getEngineOilTemperature( )
getFuelRate( )
getActualEngineTorque( )
getMILStatus( )
getEngineRunTime( )
getVIN( )
getDTC( )
clearDTC( )

All function return at least one value. The first return value is bool type, which tells us whether this function call successfully.
The other return values are truly results we want from calling this function.

4.6.1.1 getCoolantTemperature Function

getCoolantTemperature( )

Description

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

Syntax

r, temp = DFL168A.ISO15765.getCoolantTemperature( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting engine coolant temperature successfully
temp: float type, It is engine coolant temperature in Celsius degree.

4.6.1.2 getEngineSpeed Function

getEngineSpeed( )

Description

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

Syntax

r, EngineSpeed = DFL168A.ISO15765.getEngineSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting revolutions per minute of the engine crankshaft successfully

EngineSpeed: float type, It is the revolutions per minute of the engine crankshaft.

4.6.1.3 getVehicleSpeed Function

getVehicleSpeed( )

Description

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

Syntax

r, VehicleSpeed = DFL168A.ISO15765.getVehicleSpeed( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting vehicle road speed successfully

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

4.6.1.4 getIntakeManifoldPressure Function

getIntakeManifoldPressure( )

Description

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

Syntax

r, IntakeManifoldPressure = DFL168A.ISO15765.getIntakeManifoldPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting manifold pressure derived from a Manifold Absolute Pressure sensor successfully

IntakeManifoldPressure: float type, It is the manifold pressure in kPa.

4.6.1.5 getFuelSystemStatus Function

getFuelSystemStatus( )

Description

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

Syntax

r, A_Openloop, A_Closedloop, A_OpenloopByDriving_Con, A_OpenloopByFault, A_ClosedloopButFault, B_Openloop, B_Closedloop, B_OpenloopByDriving_Con, B_OpenloopByFault, B_ClosedloopButFault = DFL168A.ISO15765.getFuelSystemStatus( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting fuel system status successfully

A_Openloop: bool type, True means "fuel system 1: Open loop" , It means that it has not yet satisfied conditions to go closed loop

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

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

A_OpenloopByFault: bool type, True means "fuel system1: Open loop - due to detected system fault"

A_ClosedloopButFault: bool type, 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: bool type,True means "fuel system 2: Open loop" , It means that it has not yet satisfied conditions to go closed loop

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

B_OpenloopByDriving_Con: bool type, True means "fuel system2: Open loop due to driving conditions (e.g.power enrichment, deceleration enleanment)"

B_OpenloopByFault: bool type, True means "fuel system2: Open loop - due to detected system fault"

B_ClosedloopButFault: bool type, True means "fuel system2: Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control"

4.6.1.6 getCalculatedLoadValue Function

getCalculatedLoadValue( )

Description

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

Syntax

r, CalculatedLoad = DFL168A.ISO15765.getCalculatedLoadValue( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting calculated LOAD Value successfully

CalculatedLoad, float type, It is the calculated LOAD Value in percentage.

4.6.1.7 getShortTermFuelTrimBank13 Function

getShortTermFuelTrimBank13( )

Description

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

Syntax

r, Bank1, Bank3 = DFL168A.ISO15765.getShortTermFuelTrimBank13( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting short term fuel trim successfully

Bank1: float type, It is the short term fuel trim - Bank 1 in percentage.
Bank3:float type, It is the short term fuel trim - Bank 3 in percentage.

4.6.1.8 getLongTermFuelTrimBank13 Function

getLongTermFuelTrimBank13( )

Description

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

Syntax

r, Bank1, Bank3 = DFL168A.ISO15765.getLongTermFuelTrimBank13( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting long term fuel trim successfully

Bank1: float type, It is the long term fuel trim - Bank 1 in percentage.
Bank3: float type, It is the long term fuel trim - Bank 3 in percentage.

4.6.1.9 getShortTermFuelTrimBank24 Function

getShortTermFuelTrimBank24( )

Description

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

Syntax

r, Bank2, Bank4 = DFL168A.ISO15765.getShortTermFuelTrimBank24( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting short term fuel trim successfully

Bank2: float type, It is the short term fuel trim - Bank 2 in percentage.
Bank4: float type, It is the short term fuel trim - Bank 4 in percentage.

4.6.1.10 getLongTermFuelTrimBank24 Function

getLongTermFuelTrimBank24( )

Description

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

Syntax

r, Bank2, Bank4 = DFL168A.ISO15765.getLongTermFuelTrimBank24( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting long term fuel trim successfully

Bank2: float type, It is the long term fuel trim - Bank 2 in percentage.
Bank4: float type, It is the long term fuel trim - Bank 4 in percentage.

4.6.1.11 getIgnitionTimingAdvance Function

getIgnitionTimingAdvance( )

Description

getIgnitionTimingAdvance( ) 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( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting Ignition timing spark advance successfully

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

4.6.1.12 getIntakeAirTemperature Function

getIntakeAirTemperature( )

Description

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

Syntax

r, temp = DFL168A.ISO15765.getIntakeAirTemperature( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting intake manifold air temperature successfully

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

4.6.1.13 getAirFlowRateFrmMAF Function

getAirFlowRateFrmMAF( )

Description

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

Syntax

r, FlowRate = DFL168A.ISO15765.getAirFlowRateFrmMAF( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting air flow rate from mass air flow sensor successfully

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

4.6.1.14 getAbsThrottlePosition Function

bool getAbsThrottlePosition( )

Description

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

Syntax

r, Percent = DFL168A.ISO15765.getAbsThrottlePosition( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting absolute throttle position successfully

Percent: float type, It is the absolute throttle position in percentage.

4.6.1.15 getOxygenSensorLocation Function

getOxygenSensorLocation( )

Description

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

Syntax

r, Bank1_Sensor1Present, Bank1_Sensor2Present, Bank1_Sensor3Present, Bank1_Sensor4Present, \
Bank3_Sensor1Present, Bank3_Sensor2Present, Bank2_Sensor1Present, Bank2_Sensor2Present, \
Bank2_Sensor3Present, Bank2_Sensor4Present, Bank4_Sensor1Present, Bank4_Sensor2Present = DFL168A.ISO15765.bool getOxygenSensorLocation( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting location of oxygen sensors successfully

Bank1_Sensor1Present: bool type,True mans Bank 1 - Sensor 1 present at that location.

Bank1_Sensor2Present: bool type, True mans Bank 1 - Sensor 2 present at that location.

Bank1_Sensor3Present: bool type, True mans Bank 1 - Sensor 3 present at that location.

Bank1_Sensor4Present: bool type, True mans Bank 1 - Sensor 4 present at that location.

Bank3_Sensor1Present: bool type, True mans Bank 3 - Sensor 1 present at that location.

Bank3_Sensor2Present: bool type, True mans Bank 3 - Sensor 2 present at that location.

Bank2_Sensor1Present: bool type, True mans Bank 2 - Sensor 1 present at that location.

Bank2_Sensor2Present: bool type, True mans Bank 2 - Sensor 2 present at that location.

Bank2_Sensor3Present: bool type, True mans Bank 2 - Sensor 3 present at that location.

Bank2_Sensor4Present: bool type, True mans Bank 2 - Sensor 4 present at that location.

Bank4_Sensor1Present: bool type, True mans Bank 4 - Sensor 1 present at that location.

Bank4_Sensor2Present: bool type, True mans Bank 4 - Sensor 2 present at that location.

4.6.1.16 getBank1OSensor1Voltage Function

getBank1OSensor1Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank1OSensor1Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 1 – Sensor 1 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 1 – Sensor 1

4.6.1.17 getBank1OSensor2Voltage Function

getBank1OSensor2Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank1OSensor2Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 1 – Sensor 2 successfully.

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

4.6.1.18 getBank1OSensor3Voltage Function

getBank1OSensor3Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank1OSensor3Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 1 – Sensor 3 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 1 – Sensor 3

4.6.1.19 getBank1OSensor4Voltage Function

getBank1OSensor4Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank1OSensor4Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 1 – Sensor 4 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 1 – Sensor 4

4.6.1.20 getBank2OSensor1Voltage Function

getBank2OSensor1Voltage( )

Description

getBank2OSensor1Voltage( ) will get 0 to 1 volt oxygen sensor for Bank 2 – Sensor 1. It will return true if success, otherwise return false.

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank2OSensor1Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 2 – Sensor 1 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 2 – Sensor 1

4.6.1.21 getBank2OSensor2Voltage Function

getBank2OSensor2Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank2OSensor2Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 2 – Sensor 2 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 2 – Sensor 2

4.6.1.22 getBank2OSensor3Voltage Function

getBank2OSensor3Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank2OSensor3Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 2 – Sensor 3 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 2 – Sensor 3

4.6.1.23 getBank2OSensor4Voltage Function

getBank2OSensor4Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank2OSensor4Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 2 – Sensor 4 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 2 – Sensor 4

4.6.1.24 getBank3OSensor1Voltage Function

getBank3OSensor1Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank3OSensor1Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 3 – Sensor 1 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 3 – Sensor 1

4.6.1.25 getBank3OSensor2Voltage Function

getBank3OSensor2Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank3OSensor2Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 3 – Sensor 2 successfully.
OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 3 – Sensor 2

4.6.1.26 getBank4OSensor1Voltage Function

getBank4OSensor1Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank4OSensor1Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 4 – Sensor 1 successfully.
OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 4 – Sensor 1

4.6.1.27 getBank4OSensor2Voltage Function

getBank4OSensor2Voltage( )

Description

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

Syntax

r, OutVoltage = DFL168A.ISO15765.getBank4OSensor2Voltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting 0 to 1 volt oxygen sensor for Bank 4 – Sensor 2 successfully.

OutVoltage: float type, It is the voltage value of oxygen sensor for Bank 4 – Sensor 2

4.6.1.28 getOBDCertified Function

getOBDCertified( )

Description

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

Syntax

r, OBD = DFL168A.ISO15765.getOBDCertified( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting OBD requirements String to which vehicle or engine is certified successfully.

OBD: String type, It is the OBD certified String

4.6.1.29 getTimeSinceEngineStart Function

getTimeSinceEngineStart( )

Description

getTimeSinceEngineStart( ) 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

r, TotalTime = DFL168A.ISO15765.getTimeSinceEngineStart( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting time Since Engine Start in seconds successfully.

TotalTime: int type, It is the time (Seconds) since engine start.

4.6.1.30 getDistanceTraveledMIL Function

getDistanceTraveledMIL( )

Description

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

Syntax

r, Distance = DFL168A.ISO15765.getDistanceTraveledMIL( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the distance (Km) traveled while MIL is activated successfully.

Distance: int type, It is the distance (Km) traveled while MIL is activated

4.6.1.31 getFuelRailPressure Function

getFuelRailPressure( )

Description

getFuelRailPressure( ) 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

r, Pressure = DFL168A.ISO15765.getFuelRailPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the fuel rail pressure (kPa) at the engine successfully.

Pressure: float type, It is the fuel rail pressure in kPa.

4.6.1.32 getFuelLevelInput Function

getFuelLevelInput( )

Description

getFuelLevelInput( ) 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

r, Percent = DFL168A.ISO15765.getFuelLevelInput( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the nominal fuel tank liquid fill capacity as a percent of maximum successfully.

Percent: float type, It is the fuel level input in percentage.

4.6.1.33 getDistanceTraveledSinceDTC_Clear Function

getDistanceTraveledSinceDTC_Clear( )

Description

getDistanceTraveledSinceDTC_Clear( ) 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

r, Distance = DFL168A.ISO15765.getDistanceTraveledSinceDTC_Clear( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the distance (Km) accumulated since DTCs were cleared successfully.

Distance:int type, It is the distance (Km) traveled since DTCs cleared

4.6.1.34 getBarometricPressure Function

getBarometricPressure( )

Description

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

Syntax

r, Pressure = DFL168A.ISO15765.getBarometricPressure( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the barometric pressure successfully.

Pressure: float type, It is the barometric pressure in kPa.

4.6.1.35 getControlModuleVoltage Function

getControlModuleVoltage( )

Description

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

Syntax

r, Voltage = DFL168A.ISO15765.getControlModuleVoltage( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the control module voltage successfully.

Voltage: float type, It is the control module voltage.

4.6.1.36 getRelativeThrottlePosition Function

getRelativeThrottlePosition( )

Description

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

Syntax

r, Percent = DFL168A.ISO15765.getRelativeThrottlePosition( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the relative throttle position successfully.

Percent: float type, It is the relative throttle position in percentage.

4.6.1.37 getAmbientTemp Function

getAmbientTemp( )

Description

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

Syntax

r, AmbientTemp = DFL168A.ISO15765.getAmbientTemp( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the ambient air temperature successfully.

AmbientTemp: int type, It is the ambient air temperature in Celsius degree.

4.6.1.38 getCommandedThrottleActuatorControl Function

getCommandedThrottleActuatorControl( )

Description

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

Syntax

r, Percent = DFL168A.ISO15765.getCommandedThrottleActuatorControl( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the commanded throttle actuator control successfully.

Percent: float type, It is the commanded throttle actuator control in percentage.

4.6.1.39 getEngineRunTimeMIL Function

getEngineRunTimeMIL( )

Description

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

Syntax

r, TotalTime = DFL168A.ISO15765.getEngineRunTimeMIL( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the engine run time (minutes) while MIL is activated successfully.

TotalTime: int type, It is the engine run time (minutes) while MIL is activated.

4.6.1.40 getEngineRunTimeSinceDTC_Clear Function

getEngineRunTimeSinceDTC_Clear( )

Description

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

Syntax

r, TotalTime = DFL168A.ISO15765.getEngineRunTimeSinceDTC_Clear( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the engine run time (minutes) since DTCs cleared successfully.
TotalTime: int type, It is the engine run time (minutes) since DTCs cleared.

4.6.1.41 getTypeOfFuelUsedCurrently Function

getTypeOfFuelUsedCurrently( )

Description

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

Syntax

r, FuelType = DFL168A.ISO15765.getTypeOfFuelUsedCurrently( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the type of fuel currently being utilized by the vehicle successfully.
FuelType: String type, It is the type of fuel currently being utilized by the vehicle.

4.6.1.42 getRelativeAcceleratorPedalPosition Function

getRelativeAcceleratorPedalPosition( )

Description

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

Syntax

r, Percent = DFL168A.ISO15765.getRelativeAcceleratorPedalPosition( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the relative accelerator pedal position successfully.

Percent: float type, It is the relative accelerator pedal position in percentage.

4.6.1.43 getHybridBatteryPackRemainingLife Function

getHybridBatteryPackRemainingLife( )

Description

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

Syntax

r, Percent = DFL168A.ISO15765.getHybridBatteryPackRemainingLife( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the percent remaining life for the hybrid battery pack successfully.

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

4.6.1.44 getEngineOilTemperature Function

getEngineOilTemperature( )

Description

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

Syntax

r, Tem = DFL168A.ISO15765.getEngineOilTemperature( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the engine oil temperature successfully.

Tem: int type, It is the engine oil temperature in Celsius degree.

4.6.1.45 getFuelRate Function

getFuelRate( )

Description

getFuelRate( ) 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

r, FuelRate = DFL168A.ISO15765.getFuelRate( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the amount of fuel consumed by engine successfully.

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

4.6.1.46 getActualEngineTorque Function

getActualEngineTorque( )

Description

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

Syntax

r, ActualEngineTorque = DFL168A.ISO15765.getActualEngineTorque( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the actual engine - percent torque successfully.

ActualEngineTorque: int type, It is the actual engine - percent torque.

4.6.1.47 getMILStatus Function

getMILStatus( )

Description

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

Syntax

r, MIL_IS_ON = DFL168A.ISO15765.getMILStatus( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the Malfunction Indicator Lamp (MIL) status successfully.

MIL_IS_ON: bool type, True means MIL is ON

4.6.1.48 getEngineRunTime Function

getEngineRunTime( );

Description

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

Syntax

r, TotalEngineRunTime, TotalIdleRunTime, TotalRunTimeWithPTOActive = DFL168A.ISO15765.getEngineRunTime( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the engine run time successfully.

TotalEngineRunTime: int type, 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: int type, 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: int type, 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..

4.6.1.49 getVIN Function

getVIN( )

Description

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

Syntax

r, VIN = DFL168A.ISO15765.getVIN( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting the Vehicle Identification Number successfully.

VIN: String type, It is the Vehicle Identification Number.

4.6.1.50 getDTC Function

getDTC( )

Description

getDTC( ) will get all DTCs. It will return true if success, otherwise return false.

Syntax

r, DTC_Num, DTC = DFL168A.ISO15765.getDTC( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting all DTCs successfully.

DTC_Num: int type, It is the quantity of DTC.
DTC: list type, All elements are String type. Every element is the String of DTC.

4.6.1.51 clearDTC Function

clearDTC()

Description

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

Syntax

r = DFL168A.ISO15765.clearDTC()

Parameters

Nothing

Returns

r : bool, Ture denotes clearing all DTCs successfully.

4.7 Sub-module GPS

4.7.1 Function

getGPSinfo( )

getAltitude( )

All function return at least one value. The first return value is bool type, which tells us whether this function call successfully.

The other return values are truly results we want from calling this function.

4.7.1.1 getGPSinfo Function

getGPSinfo( )

Description

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

Syntax

r, Latitude, Longitude, Speed, Time, Date = DFL168A.ISO15765.getGPSinfo( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting GPS location and date and time information successfully.

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

Longitude: float type, It is the Longitude of vehicle in degree. Plus is east, Minus is west.

Speed: float type, It is the vehicle speed based on GPS navigation in Km/h.

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

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

4.7.1.2 getAltitude Function

getAltitude( )

Description

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

Syntax

r, Altitude = DFL168A.ISO15765.getAltitude( )

Parameters

Nothing

Returns

r : bool, Ture denotes getting altitude of vehicle location successfully.

Altitude: float type, It is the altitude of vehicle in meter.

5 Examples

All examples are in `https://github.com/Dafulai/DFL168A_python/examples`.

You just modify your serial port Name to your actual Serial port such as "COM1" (or "/dev/ttyS0") instead of default "COM4". You can run it directly.