DFL168A Rev 1.10 Datasheet

Table of Contents

I Overview 4

II Features Highlights 4

III Typical Application 5

IV Pinout 5

V Reference Schematics 9

VI Communicate with DFL168A 10

1 Overview ... 10
2 Quick Start ... 11
3 AT Command Summary ... 14
4 Vehicle BUS Access ... 20
- J1939 ... 20
- J1708/J1587 ... 24
- ISO15765 ... 26
- USER1/USER2 ... 30

5 Device 1 Access ... 32
- General Device with Binary response ... 32
- General Device with ASCII response ... 35
- CS440 controller ... 37
- CS230 Controller ... 38
- Dickey-John Controller ... 39
- ACE Spreader Controller ... 41
- GPS with NMEA 0183 output ... 43
- Transparent mode ... 47

6 Power Management ... 48
7 IC Self-Diagnosis ... 50
8 Higher Baud Rate for UART1 ... 52
9 AT Command Details ... 54
10 Programmable Parameters Information ... 76
11 ERROR and Warning Information ... 80

VII Electrical Characteristics 83

VIII Packaging Diagrams and Parameters 85

1 Overview

DFL168A is an IC, which can access SAE J1939, SAE J1708, ISO15765 and CAN Bus user protocols, and lots of spreader or GPS serial device or one-wire device or inputs and output by an interface UART port. So One UART port can access Heavy Duty Vehicle data and Spreader data and GPS data and one-wire data and discrete inputs/output and analog input in real time. It will be perfect for fleet management system.

We know lots of AVL (Automatic Vehicle Locator) in the market, which only has one RS232 interface. However, As for an fleet management system, we need to monitor not only GPS location, but also vehicle and equipment parameters on the vehicle such as spreader.

DFL168A provides a good solution. You only need one RS232 or UART, and you can monitor vehicle and spreader status and i-button or other discrete/Analog inputs.

2 Features Highlights

- One interface with baud rate from 9600 to 5000KBPS*
- Support SAE J1939/FMS, SAE J1708/J1587 Heavy Duty
- Support ISO15765 and CAN bus user protocols
- Support the following Devices
  1. General Binary Device
  2. General ASCII Device
  3. CS440 controller
  4. CS230 Controller
  5. Dickey-John Controller
  6. ACE Spreader Controller
  7. GPS with NMEA 0183 output
- Support transparent mode for device
- Support 4 Discrete inputs and one Discrete output and one analog input
- Contains one-wire Master interface, which supports one-wire slave I-button.
- Vehicle Speed PWM output and Forwad/Backward output, It can used for spreader controller as speed input or It will make GPS DR function easier.
- Truck Data Bus and Spreader/GPS can be worked at the same time
- Support Self-Diagnose function
- Support Sleep mode and non-ignition wake-up method, which makes wiring easier.
- Support non-intrude command which makes IC not to intrude any signals into Truck Data Bus
- Available in 28 pins SPDIP, SOIC and QFN packages
- RoHS Compliant

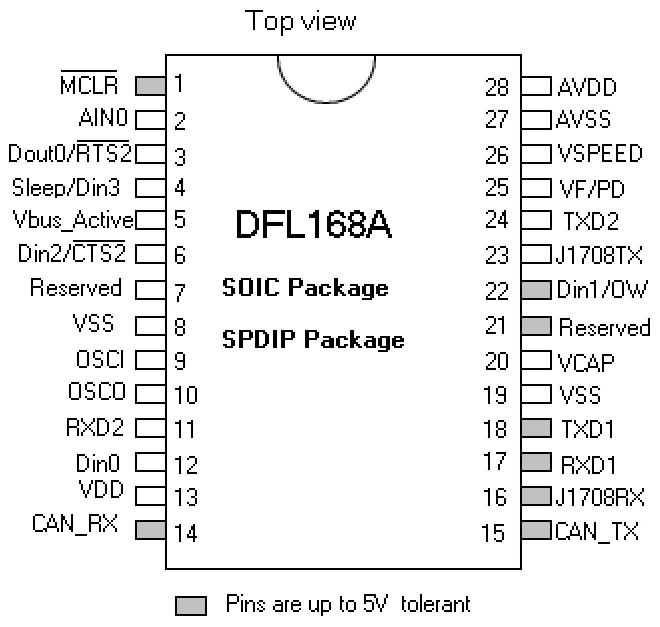
Note: **Actual Baud rate will depend on hardware*

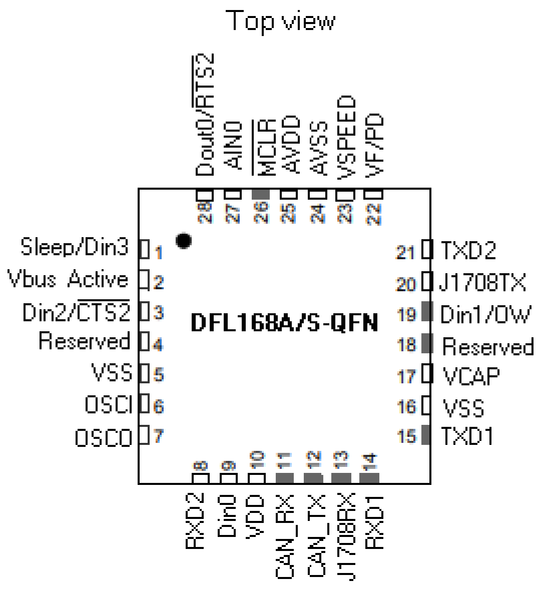
3 Typical Application

- Fleet Management and Tracking Device
- Automotive diagnostic scan tools and code readers
- Sensor with J1939/J1708/ISO15765 and CAN bus protocol
- Digital dashboards

4 Pinout

DFL168A has 28 pins SPDIP and SOIC and QFN-S packages





■ Pins are up to 5V tolerant

The exposed copper pin in the center is recommended to be connected to VSS

Pin Descriptions:

- **/MCLR (SOIC and SPDIP: Pin 1, QFN: Pin26)**

  A reset Input pin. Logic 0 longer than 2uS will bring IC to reset state. If you don't use it, use 10K ohms' resistor to pull up to VDD.

- **Ain0 (SOIC and SPDIP: Pin 2, QFN: Pin27)**

  Analog input pin. The analog voltage range is 0.0 to AVDD.
  If it is 0.0V when IC restarts, IC will use the default baud rate (57600bps) for UART1

- **Dout0/RTS2 (SOIC and SPDIP: Pin 3, QFN: Pin28)**

  Discrete Output 0 and /RTS2 of UART2. It is a digital output port. Factory default is digital output port. Device 1 is an external device using UART2 interface. You can use command "AT DEV1 HFC 0" to disable RTS2 function. When RTS2 is disabled, this pin will be Dout0. You can use command "AT DEV1 HFC 1" to enable /RTS2 function. /RTS2 is output of hardware flow control for UART2. This is active low output. UART2 connects external Device 1.

- **Sleep/Din3 (SOIC and SPDIP: Pin 4, QFN: Pin1)**

  Sleep Input and Discrete Input Din3 pin. Default is Din3. If you change to sleep input by command "AT Sleep pin 1", the default will be active low.

- **Vbus_Active (SOIC and SPDIP: Pin 5, QFN: Pin2)**

Vehicle Bus Active output. It will be logic 1 when vehicle bus is active.

- **Din2/CTS2 (SOIC and SPDIP: Pin 6, QFN: Pin3)**
  Discrete Input 2 and /CTS2 of UART2. It is a digital input port. Factory default is digital input. /CTS2 is is input of hardware flow control for UART2. This is active low input. UART2 connects external Device 1. You can use command "AT DEV1 HFC 1" to enable RTS2 function. You can use command "AT DEV1 HFC 0" to disable /CTS2 function. When /CTS2 is disabled, This pin will be Din2.

- **Reserved (SOIC and SPDIP: Pin 7, QFN: Pin4)**
  Reserved Output pin. It is used for factory test. No connect.

- **VSS (SOIC and SPDIP: Pin 8, QFN: Pin5)**
  Ground reference for logic and discrete input/output pins.

- **OSCI (SOIC and SPDIP: Pin 9, QFN: Pin6) and OSO (SOIC and SPDIP: Pin 10, QFN: Pin7)**
  For SOIC and SPDIP package, a 4.00MHz crystal is connected between these two pins. Loading capacitors as required by the crystal (typically 27pF each) will also need to be connected between each of these pins and VSS.
  For QFN package, an 8.00MHz crystal is connected between these two pins. Loading capacitors as required by the crystal (typically 27pF each) will also need to be connected between each of these pins and VSS.
- **RXD2 (SOIC and SPDIP: Pin 11, QFN: Pin8)**
  Receive data input of UART2. UART2 connects external Device 1.

- **Din0 (SOIC and SPDIP: Pin 12, QFN: Pin9)**
  Discrete Input Din0 pin.

- **VDD (SOIC and SPDIP: Pin 13, QFN: Pin10)**
  Positive supply for peripheral logic and discrete input/output pins

- **CAN_RX (SOIC and SPDIP: Pin 14, QFN: Pin11) and CAN_TX (SOIC and SPDIP: Pin 15, QFN: Pin12)**
  Theses are interface signals of CAN BUS. You have to connect them to CAN BUS transceiver IC.

- **J1708RX (SOIC and SPDIP: Pin16, QFN: Pin13)**
  J1708 receive data pin. It connects the RS485 transceiver IC

- **RXD1 (SOIC and SPDIP: Pin 17, QFN: Pin14)**
  Receive data input of UART1. UART1 is interface port of external commands.

- **TXD1 (SOIC and SPDIP: Pin 18, QFN: Pin15)**
  Data transmit input of UART1. UART1 is interface port of external commands

- **VSS (SOIC and SPDIP: Pin 19, QFN: Pin16)**
  Ground reference for logic and discrete input/output pins.

- **VCAP (SOIC and SPDIP: Pin 20, QFN: Pin17)**
  Filter capacitor connection. A low-ESR (< 5 Ohms) capacitor is required on the VCAP pin, which is used to stabilize the voltage regulator output voltage. The VCAP pin must not be connected to VDD, and must have a capacitor between 4.7 µF and 10 µF, 16V connected to ground. The type can be ceramic or tantalum.

- **Reserved (SOIC and SPDIP: Pin 21, QFN: Pin18)**
  Reserved Input pin. It is used for factory test. No connect.

- **Din1/OW (SOIC and SPDIP: Pin 22, QFN: Pin19)**
  Discrete Input Din1 pin and One-wire Bus Pin. This is open-drain pin. Please use external pull-up resistor.

- **J1708TX (SOIC and SPDIP: Pin 23, QFN: Pin20)**
  J1708 data transmit pin. It connects the RS485 transceiver IC

- **TXD2 (SOIC and SPDIP: Pin 24, QFN: Pin21)**
  Data transmit output of UART2. UART2 connects external Device 1.

- **VF/PD (SOIC and SPDIP: Pin 25, QFN: Pin22)**
  Vehicle Forward and Power down output pin. You can change the active voltage by AT commands. You can select vehicle forward output or power down output by AT commands. The factory default is power down output. Default value is logic 1 when power down (sleep).

- **VSPEED (SOIC and SPDIP: Pin 26, QFN: Pin23)**
  Vehicle speed pulse output. The frequency of output pulse is directly proportional to vehicle speed. The scale can be changed by AT command. The default is 20Hz for each Km/h.The duty cycle of

pulse is 50%.

- **AVSS (SOIC and SPDIP: Pin 27, QFN: Pin24)**

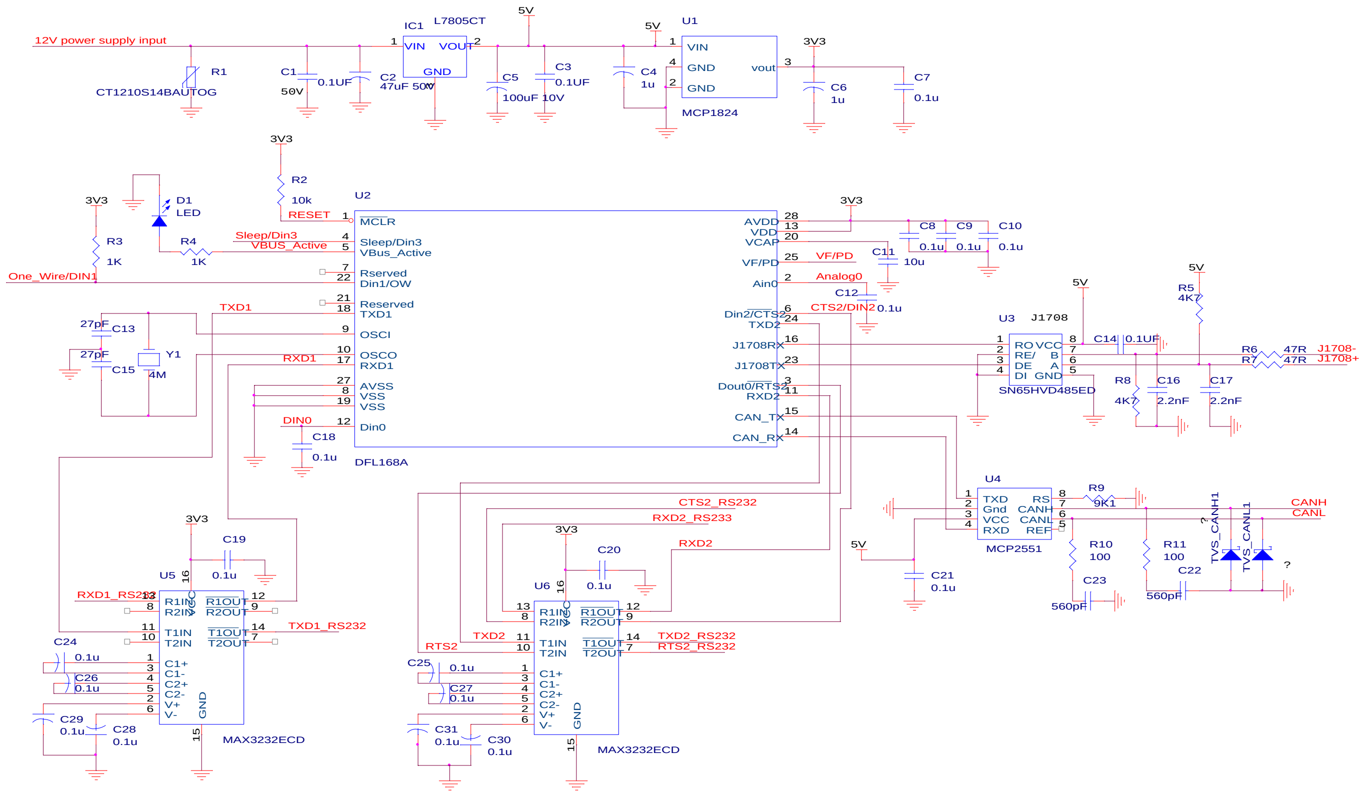
  Ground reference for analog modules.

- **AVDD (SOIC and SPDIP: Pin 28, QFN: Pin25)**

  Positive supply for analog modules. This pin must be connected at all times.

5 Reference Schematics

```
The recommended schematic for SOIC and PSDIP package is shown below:
```



Notes: *1 We didn't consider the PD pin to control transceiver MCP2551 and RS232 for the above schematic. Customers have to consider the PD pin to control some ICs to stand-by mode, and choose low quiescent current regulator..*

2 For QFN Package, Crystal Y1 must be changed to 8MHz, and U2 Pin numbers are changed, you must use QFN Pin definition.

6 Communicate with DFL168A

6.1 Overview

DFL168A IC interprets data from vehicle and external spreader to one UART. So user can use one UART port to access vehicle data, spreader data, GPS data, one-wire data, and discrete or analog input data. DFL168A is communicated with user via a UART port (UART1) , which only needs TXD and RXD line. In general, we can use hyperterminal running on PC with the setting of 57600, 8-N-1,No Flow control. Of cause, the RS232 logic level has to be changed into 3.3V TTL logic level by RS232 IC.

When DFL168A properly connected and power on, you will see the information below in the hyperterminal

```
Dafulai Electronics, DFL168A Starting

>
```

The second line ">" is a hint symbol, it means that DFL168A is ready to receive command from outside. Right now, if you type the letters A T and i (space is optional) and then press key "Enter":

```
>AT i
```

You should see the information below:

```
DFL168A V1.10

>
```

It displayed us the type of IC and revision number. Here "AT" is command header. "I" is the specific command name. In general, all commands header is "AT" . However, for commands, which directly access data from vehicle bus, there are no any "AT" header, it is directly hexadecimal digits (0 to 9 and A to F). If the command you type has syntax error or the command does not exist, you will see the question mark "?".

All commands are case-insensitive.

If you type wrong letter, you want to change it, just use "backspace" key to delete it.
If you type a letter, and pause for 10 seconds without typing any letter, you will see the question mark (?) . This is for anti-interferes. In the field, "Enter" key (Hex 0d) maybe destroyed due to noise, So it will make "DFL168A" deadly waiting for "Enter" key forever if without the 10 seconds limit.

6.2 Quick Start

For accessing J1939 PGN, just type "AT SP A" and key "Enter" to enter "J1939 protocol"

```
>AT SP A
```

You will see response:

```
OK

>
```

And then type the PGN number you request in hexadecimal digits and key "Enter", for example,

```
>F004
```

You will see 8 bytes' response in Hex below:

```
FF FF A2 00 7D FF FF FF

>
```

If you want to request other PGN, type other PGN number you request in hexadecimal digits again and you will see the hex response.

For accessing J1708/J1587, just type "AT SP D" and "Enter" to enter "J1708 protocol"

```
>AT SP D
```

You will see response:

```
OK
>
```

And then type the PID number you request in hexadecimal digits and "Enter", for example,

```
>5C
```

You will see one byte's response in Hex (Percent engine load)

```
C7
>
```

If you want to request other PID, type other PID number you request in hexadecimal digits again and you will see the hex response.

For accessing ISO15765, just type "AT SP 0" and "Enter" to enter "ISO15765" Protocol. IC will search specific ISO15765 automatically.

```
>AT SP 0
```

You will see response:

```
OK
>
SEARCHING...
>
```

And then type the service number and PID number you request in hexadecimal digits and "Enter", for example, you want to get the engine rpm, service number is 0x01 and PID number is 0x0C. So just type 010C

```
>01 0C
```

You will see

```
41 0C 1A F8
>
```

41 0C tells us service number 01 and PID number 0C for response data.
The returned value 1A F8 is hex value, convert it to decimal 6904, scale is 0.25, so engine rpm is 6904*0.25=1726 rpm.

For accessing DEV1 device, for example , Dickey-John Controller, just type "AT DEV1 SP 4 " and "Enter" to enter "Dickey-John Controller"

```
>AT DEV1 SP 4
```

You will see response:

```
OK
>
```

And then type the "AT DEV1 DJ RD" and 2 Hex digital Data group number and 2 Hex digital Data Element Number and key "Enter". For example, Data group is Hex F0, Data Element is Hex 04. This is a ground speed parameter. Type the command below

```
>AT DEV1 DJ RD F0 04
```

You will see response:

```
04 105
>
```

The first 2 digit is hex number, it is data element number you request. The number 105 followed hex 04 is decimal ground speed. It is 105 Km/h.
If you want to request other data group and data element, just type "AT DEV1 DJ RD" command again.
If you want to read discrete input, for example DIN2, just type "AT RD 2" and key "Enter"

```
>AT RD  2
```

You will see response:

```
1

>
```

It means DIN2 is logic 1.

If you want to read analog input, just type "AT RV" and key "Enter"

```
>AT RV
```

You will see response:

```
12.0V

>
```

It means analog input is 12.0V
If you want to read I-button, just type"AT OW RD" and key "Enter"

```
>AT OW RD
```

You will see response:

```
67 78 F2 E4 25 C1 01

>
```

Total 7 bytes hex response, the first 6 bytes is unique Serial number in the world, the last byte hex 01 is device family code. For I-button, it must be hex 01.
If you want to make discrete out0 (DOUT0) to be 1, just type "AT WD 0 1" and key "Enter"

```
>AT WD 0 1
```

You will see response:

```
OK

>
```

The above information is enough for beginner. If you want to know more about the operation of the IC, please continue to read.

6.3 AT Command Summary

The symbol "hh" below denotes hexadecimal digit.
The symbol "xx" below denotes hexadecimal digit.
The symbol "yy" below denotes hexadecimal digit.
The symbol "zz" below denotes hexadecimal digit.
The symbol "dddd " below denotes decimal digit "dd.dd"

General Commands:

<CR>	:repeat the last command
BRD hh	:try UART 1 Baud Rate Divisor hh
BRT hh	:set UART 1 Baud Rate Timeout
D	:set all to Defaults
E0, E1	:Echo Off, or On*
H0, H1	:response header off*, or header On
I	:print the version ID
L0, L1	:Linefeeds Off, or On*
S0, S1	:printing of space Off, or On*
SelfD 0	:start IC self-Diagnosis without discrete out0 output
SelfD 1	:start IC self-Diagnosis with discrete out0 alternating 0 and 1
Z	:reset all
@R	:reset all
@1	:display the IC description
@2	:display copyright

Programmable Parameter Commands

PP xx OFF	:disable Programmable Parameter xx
PP FF OFF	:disable all Programmable Parameter

PP xx ON	:enable Programmable Parameter xx
PP FF ON	:enable all Programmable Parameter
PP xx SV yy	:for Programmable Parameter xx, Set the value to yy
PPS	:print a Programmable Parameter Summary
PPP	:all Programmable Parameter store into flash memory

Truck Data Bus Commands

DP	:Describe the current Protocol by text
DPN	:Describe the current Protocol by Number
INTRUDE 0, INTRUDE 1	:transmit message into truck data bus Off, transmit message into truck data bus On*
MA	:Monitor All
PC	:Protocol Close
R0, R1	:Response Off, or On*
SP h	:Set current Protocol to h and save it
ST hh	:Set response Timeout to hh
TP h	:Try current Protocol to h (not save)
VF1, VF0, VF2	: Vehicle Forward pin logic 1 out, or Vehicle Forward pin logic 0 out, or Disable Vehicle Forward pin out*.

J1708/J1587 Specific Commands

TMID xx	:set Tester MID to xx
MMID 00	:Monitor all MID
MMID xx	:Monitor MID xx
MMID xx 00	:Monitor MID xx
MMID xx hh hh ... hh	:Monitor MID xx with multiple different PID hh, maximum quantity of PID is 8, minimum quantity of PID is 1

J1939 Specific Commands (Protocol A to C)

ADC0, ADC1	:Address claim off*, Address claim On
D0, D1	:Display of DLC Off*, Display of DLC On
DM1	:Monitor for DM1 messages
JB	:J1939 request PGN data use big endian*
JE	:J1939 request PGN data use big endian*
JL	:J1939 request PGN data use little endian
JS	:J1939 request PGN data use little endian

MP hhhh	:Monitor for PGN hhhh (ignore priority and page number)
MP hhhhh	:Monitor for PGN hhhhh (only ignore priority)

CAN BUS Specific Commands (Protocol 6 to 9 and protocol B and C)

AD0, AD1	:Mixed address off*, Mixed address On
ADT hh	:Set mixed address to hex hh. Default is hex 80
AR	:Automatically Receive
CAF0, CAF1	:ISO15765 Auto PCI Off, ISO15765 Auto PCI On*
CF hhh	:Set the ID Filter to hhh
CF hhhhhhhhh	:Set the ID Filter to hhhhhhhh
CFC0, CFC1	:Flow Control Off, or On*
CM hhh	:Set the ID Mask to hhh
CM hhhhhhhh	:Set the ID Mask to hhhhhhhh
CP hh	:Set CAN Priority to hh (29 bits ID CAN Bus request)
CRA hhh	:Set 11 bits of ID for CAN Bus response
CRA hhhhhhhh	:Set 29 bits of ID for CAN Bus response
D0, D1	:Display of DLC Off*, Display of DLC On
FCSD hh	:Flow control, Set Data to one byte hh
FCSD hhhh	:Flow control, Set Data to two bytes hh hh
FCSD hhhhhh	:Flow control, Set Data to three bytes hh hh hh
FCSD hhhhhhhh	:Flow control, Set Data to four bytes hh hh hh hh
FCSD hhhhhhhhhh	:Flow control, Set Data to five bytes hh hh hh hh hh
FCSM h	:Flow Control, Set the mode to h
FCSH hhh	:Flow Control, Set the 11 bits ID to hhh
FCSH hhhhhhhh	:Flow Control, Set the 29 bits ID to hhhhhhhh
MR hh	:Monitor for receiver's address hh
MT hh	:Monitor for transmitter's address hh
RTR	:Send an Remote frame
SH hhh	: Set 11 bits of ID for CAN Bus Request
SH hhhhhh	: Set lower 24 bits of ID for CAN Bus Request, High 5 bits is set by "AT CP hh"

Device1 Specific Commands

DEV1 BRD hh	:set UART 2 (Device 1) Baud Rate Divisor hh, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 BT hh	:set UART 2 (Device 1) Data packet Broken Time hh, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 CLR	:Clear the history data of UART 2 (Device 1).
DEV1 CS h	:set Check Sum adding extra h, Must execute under Dev1 protocol is

	close by "AT DEV1 PC"
DEV1 CS hh	:set Check Sum adding extra hh, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 CS hhhh	:set Check Sum adding extra hhhh, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 CSM hh	:set Check Sum Mode hh, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 DP	:Describe device1 Protocol by text
DEV1 DPN	:Describe device1 Protocol by Number
DEV1 EOF	:cancel the "End of Frame" byte for Device 1, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 EOF hh	:set the "End of Frame" byte to hh for Device 1, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 Exit T hh	: set DEV1 exiting transparent mode character to hh.
DEV1 HFC 1, DEV1 HFC 0	:enable hardware flow control RTS2/CTS2, or disable hardware flow control RTS2/CTS2, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 HT 1, DEV1 HT 0	:enable history data of UART 2 (Device 1), or disable history data of UART 2 (Device 1).
DEV1 GB	:Get Broadcast data of Device 1, don't care which SOF it is
DEV1 GB SOF h	:Get Broadcast data of Device 1 with the SOF number h
DEV1 LEN	:cancel the receiving data packet length of device 1, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 LEN hh	:set the receiving data packet length of device 1 to fixed value hh, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 LEN xx hh	:set the receiving data packet length of device 1 to a value, which is the byte located in xx position of receiving data packet and we must add hh to the value, Must execute under Dev1 protocol is close by "AT DEV1 PC". Please read details.
DEV1 REQ hh hh ... hh	:transmit Request hh hh ...hh to Device 1. The meaning is different for ASCII Device and Binary Device
DEV1 RSOF h	:set Response of Device1 with SOF number h
DEV1 SI GB	:Get Broadcast data of Device 1, don't care which SOF it is. The response does not display, it's called "silence". Response data is only used by internal data analysis.
DEV1 SI GB SOF h	:Get Broadcast data of Device 1 with SOF number h. The response does not display, it's called "silence". Response data is only used by internal data analysis.
DEV1 SI REQ hh hh ... hh	:It is the same as "DEV1 REQ hh hh ... hh" except "Silence"

DEV1 S	:cancel the separating time between the receiving data packet, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 S hh	:set the separating time hh between the receiving data packet, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h	:cancel the SOF of Number h, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h xx	:set the SOF of Number h to xx, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h xx xx	:set the SOF of Number h to xx xx, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h xx xx xx	:set the SOF of Number h to xx xx xx, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h xx xx xx xx	:set the SOF of Number h to xx xx xx xx, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h xx xx xx xx xx	:set the SOF of Number h to xx xx xx xx xx, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SOF h xx xx xx xx xx xx	:set the SOF of Number h to xx xx xx xx xx xx, Must execute under Dev1 protocol is close by "AT DEV1 PC"
DEV1 SP h	:set Device 1 Protocol h and save it
DEV1 ST hh	:Set Device 1 response Timeout to hh
DEV1 TP h	:set Device 1 Protocol h and no save it
DEV1 TPK 1, DEV1 TPK 0	:Enable (1*)/Disable(0) Device 1 transparent mode exiting key character.
DEV1 TPT 1, DEV1 TPT 0	:Enable (1*)/Disable(0) Device 1 transparent mode time out.
DEV1 TRANSP	: Let Device 1 enter "Transparent mode"
DEV1 PC	:Device 1 Protocol Close

CS440 Controller Specific Commands

DEV1 REQ	:send data Request to CS440 controller of device 1
DEV1 SI REQ	:It is the same as "DEV1 REQ " except "Silence"

CS230 Controller Specific Commands

DEV1 ADR REQ hhhh	:send data Request to CS230 controller of device 1 in order to get data of Address hhhh

Dickey-John Controller Specific Commands

DEV1 DJ RD xx yy	:Dickey-John controller (Device 1) Read data from data group xx and

data element yy

DEV1 DJ RD xx yy zz	:Dickey-John controller (Device 1) Read data from data group xx and data element yy and data sub element zz

ACE Spreader Controller Specific Commands

DEV1 AC1	:get AC1 Broadcast data of ACE controller (Device 1)
DEV1 AC2	:get AC2 Broadcast data of ACE controller (Device 1)
DEV1 AC3	:get AC3 Broadcast data of ACE controller (Device 1)
DEV1 ACE DIST	:get the Distance traveled of ACE controller (Device 1)
DEV1 ACE DIST hhhh	:set the initial Distance traveled to hhhh
DEV1 ACE QtyOfS	:get the spread Quantity of Solid material from ACE controller (Device 1)
DEV1 ACE QtyOfS hhhh	:set the initial spread Quantity of Solid material to hhhh
DEV1 ACE QtyOfL	:get the spread Quantity of Liquid from ACE controller
DEV1 ACE QtyOfL hhhh	:set the initial spread Quantity of Liquid to hhhh

GPS Specific Commands

DEV1 Alt	:get Altitude from GPS (Device 1)
DEV1 Date	:get Date from GPS (Device 1)
DEV1 Lat	:get Latitude from GPS (Device 1)
DEV1 Long	:get Longitude from GPS (Device 1)
DEV1 Speed	:get Speed from GPS (Device 1)
DEV1 Time	:get Time from GPS (Device 1)

Power Management Specific Commands

PC	:Truck Protocol Close. It may cause sleep when sleep pin is not enabled
SLEEP hhhh	:set Sleep delay time to hhhh seconds
SLEEP P 1, SLEEP P 0	:Sleep pin Polarity is logic-1 active, or Sleep pin Polarity is logic-0 active
SLEEP PIN 1, SLEEP PIN 0	:Sleep Pin enable, or Sleep Pin disable
PD1, PD0, PD2	:Power Down pin out logic1*, or Power Down pin out logic 0, or Disable Power Down pin out.

Input/output Specific Commands

CV 0000	:Calibrate the voltage to default.
CV dddd	:Calibrate the voltage to dd.dd (Decimals)
DEBOUNCE 1, DEBOUNCE 0	:enable switch debounce, or disable switch debounce
RD h	:Read discrete input h (Din h)
RV	:Read analog input (value is decimals)

WD 0 1, WD 0 0	:Write Dout0 1, or Write Dout0 0

One-wire Device Specific Commands

OW RD	:One-Wire Read

Note: ** denotes default status*

6.4 Vehicle BUS Access

DFL168A supports heavy-duty vehicle bus protocol: SAE J1939 and J1708/J1587, and middle or light duty vehicle bus protocol ISO15765 and Customized CAN bus protocols(User1 and User2) . Customers only need to send digital request, they can easily get digital responses.

6.4.1 J1939

The SAE J1939 CAN standard is a protocols used in heavy duty vehicle. It uses ISO 11898 standard CAN physical interface, and defines its data format and transfer. In DFL168A, J1939 Protocol number is A, you can use "AT SP A" (stored your selected protocol) or "AT TP A" (Not stored your selected protocol) to select J1939.

DFL168A provide the functions below for J1939 Protocol:

1. Send an address claim before using its own source address (option)
2. After receiving address claim, DFL168A can send address claimed message or cannot claim message
3. Eight bytes' name can be programmed in PP 20 to 30
4. Source Address can be changed by PP 1E, and can be automatically changed in address arbitration process
5. Support long messages's BAM TP.DT and RTS/CTS.
6. Baud rate can be changed by PP 32
7. Automatically output vehicle speed pulse and vehicle forward/backward logic level, and frequency can be scaled by PP 1D, forward/backward logic level can be changed by PP 1A

J1939 baud rate is 250Kbps according to SAE J1939-11. However, you can change to other baud rate by PP 32.

Firstly you have to make sure current protocol is J1939. You can get current protocol by "AT DP" command. If current protocol is not J1939, please use "AT SP A" or "AT TP A" to change it. And then when you send a request, you only need to type the PGN you request. DFL168A automatically add some necessary header to produce request PGN. For example, you want ECU Engine temperature parameter, J1939 will use PGN 65262 (Hex: 00FEEE) to transfer this message. You just send 00FEEE to UART1:

```
>00FEEE
```

If you don't care page number, you can send FEEE to UART1:

```
>FEEE
```

If you didn't use the default big-endian, you will send EEFE00 or EEFE to UART1

```
>EEFE00
```

or

```
>EEFE
```

You will see 8 bytes' response from UART1 if header is off:

```
BC 78 3F BC FF FF FF FF

>
```

You will see header and DLC and 8 bytes' response from UART1 if header is on (AT H1) and DLC is on (AT D1):

```
6 0FEEE 00 8 BC 78 3F BC FF FF FF FF
```

You will see header and 8 bytes' response from UART1 if header is on (AT H1) and DLC is off (AT D0):

```
6 0FEEE 00 BC 78 3F BC FF FF FF FF
```

The display format for single frame message is:

1 Hex(Priority) space 1 Hex (Reserve bit and Page number) space 2 Hex (PF) 2Hex(PS) space 2 Hex (Source Address) Space 1 Hex (DLC if AT D1) space 2 Hex (Data) space 2 Hex (Data) space ... 2 Hex (Data)

At most 8 bytes' data occurs in the format above, and DLC occurs only when AT D1, and space occurs only when AT S1. So the display above tells us Priority is 6, Reserve bit and page number is 0, PF is FE, PS is EE, Source address is 00, DLC is 8, Data bytes are BC 78 3F BC FF FF FF FF

DFL168A can handle multi-frame response message, and it will prepare correct data response for transportation. You don't need to know how to run for multi-packet. DFL168A does everything for you. You only need to see data in UART1.

The display format for multi-packets when header is off is shown below:

3 Hex (Total bytes)

2 Hex(Frame serial number) colon(:) space 1 Hex (DLC if AT D1) 2 Hex (Data) space ... 2 Hex (Data) (Total 7 bytes ' data)

....

2 Hex(Frame serial number) colon(:) space 1 Hex (DLC if AT D1) 2 Hex (Data) space ... 2 Hex (Data) (Maybe < 7 bytes)

The display format for multi-packets when header is on is almost the same as one of single packet except that there are 3 Hex to denote total bytes in the first line.

For example, we request PGN DM1 65226 (00FECA) when 2 faults happened . We type the following command

```
>ATH0
OK

>FECA
00A
01: 43 FF B8 04 03 8A 90
02: 02 03 82 FF FF FF FF

>ATH1
OK

>ATD1
OK

>FECA
00A
7 0EBFF 00 8 01 43 FF B8 04 03 8A 90
7 0EBFF 00 8 02 02 03 82 FF FF FF FF

>
```

Sometimes, we maybe get "No Data" response, we can increase command time out value by AT ST hh command. If DFL168A didn't return ">" for long time because of waiting for vehicle response, you can use "Ctrl+C" to pause the waiting and return ">", you can execute some non-vehicle data buss access command. Afterwards, DFL168A can return data response from vehicle when data is ready.

DFL168A can enter monitor mode by typing command "AT MA" or "AT DM1" or "AT MP hhhh" or "AT MP hhhhh" in hyperterminal.

DFL168A will continuously display the PGN which it monitored. Pressing any key in the hyperterminal will exit monitor mode and return ">" normal state. Display in the monitor mode is the same as one of normal state. However, "AT MA" command of J1939 will display header bytes no matter whether header is on. This reason is that all PGNs information is contained in CAN ID. If you can't see header in different PGNs, you will feel the data received to be no sense.

"AT DM1" will monitor PGN 65226.
"AT MP hhhh" will monitor PGN hhhh and it will ignore page number and priority .
"AT MP hhhhh" will monitor PGN hhhh and it will ignore priority.
"AT JB", "AT JE", "AT JL",and "AT JS" will affect PGN byte order of "AT MP hhhh" and "AT MP hhhhh" command.

We have introduced the 2 and 3 bytes' request, they are PGN requests with global address.

However, there are other messages which are more than 3 bytes to be send to truck data bus. These are any J1939 Messages. We take the following format to send:

1 Hex (Priority) 1 Hex (Reserve and Page number) 2 Hex (PF) 2Hex (PS) 2 Hex(DLC) 2 or 4 or 6 or 8 or 10 or 12 or 14 or 16 Hex (Data Bytes)

Data byte order will be little-endian , it will bypass "AT JB" and "AT JE" command.
For example,

```
> 6 0 FE E9 08 EE FE 00 01 45 67 00 00
```

will be PGN 65257 (FE E9) broadcast. In this way, we can send any PGN to truck data bus. We can make a simple J1939 simulator, following the method below:

Firstly, monitor PGN 59904 (EA00)

```
>AT MP 00EA00
```

Secondly, when we got the PGN 59904, we can get the request PGN from its data field. And then send the PGN with simulated data by digital command which has more than 3 bytes.

Of cause, you can make a J1939 Analog/Digital sensor. Firstly you use "AT RD h" or "AT RV" command to get sensor data (you can use UART2 as DEV1 to get sensor data too), and then you can send PGN based on sensor data to CAN BUS.

Notes: *1 You have to fully understand the J1939 protocol if you want to send message longer than 3 bytes, otherwise you may damage the vehicle*

2 If you send PGN 59904 by digital command longer than 3 bytes, you will get PGN response. In this way, You can send PGN request with the local destination address. The PGN request has global destination address for 3 bytes' or 2 bytes' command.

Furthermore, J1939 broadcast most of messages periodically. In most of time, we don't need actually send a request to truck data bus. The command "AT INTRUDE 0" will make DFL168A not to send any message to truck data bus. But 3 bytes or 2 bytes command can still get response if there is a broadcast for the request PGN. This way makes DFL168A no any intruder to truck data bus.

Vehicle speed pulse output and vehicle forward/backward output are spontaneous, you don't need to send PGN 65132 because PGN 65132 broadcasts every 50 ms.

6.4.2 J1708/J1587

The SAE J1708/J1587 standard is a protocols used in heavy duty vehicle. J1708 is a simple multi-master bus interface. The specification describes a physical layer based upon an RS-485 bus. By using a recessive state and a dominate state for the bus,multiple masters can share the transport media without fear of contention. The J1708 protocol includes methods for claiming the bus, resolving collisions, and transmitting data. The data is transmitted at 9600 baud with 1 Start bit, 8 data bits and 1 Stop bit. The packet length is 2 to 21 bytes including a checksum for error detection. Parameter Identification (PID) numbers identify the data on the bus.
In DFL168A, J1708/J1587 Protocol number is D, you can use "AT SP D" (stored your selected protocol) or "AT TP D" (Not stored your selected protocol) to select J1708/J1587.

SAE J1587 is a specification which defines messages that are transmitted on a SAE J1708 network. J1708 specifies the data link and physical layers, while J1587 specifies the transport, network, and application layers.
All messages have the following format:

MID, PID/Data, [PID/Data, PID/Data, ...], Checksum

Messages start with a MID, which stands for message identifier and indicates the source address of the transmitting node.

DFL168A provide the functions below for J1708/J1587 Protocol:

1. MID can be changed by PP 06 permanently, or changed by "AT TMID hh" temporally.
2. Multisection parameter PID 192 can be combined into a message longer than 21 bytes to display
3. Automatically output vehicle speed pulse, and frequency can be scaled by PP 1D

Firstly you have to make sure current protocol is J1708. You can get current protocol by "AT DP" command. If current protocol is not J1708, please use "AT SP D" or "AT TP D" to change it. And then when you send a request, you only need to type one byte's PID number you request. For example, you want engine speed which PID is 190 (Dec) or BE (Hex), just type BE:

```
>BE
BE E0 2E

>
```

The response is BE E0 2E. The first byte is PID, So the engine speed will be 2EE0 because J1708/ J1587 is little-endian. Bit resolution is 0.25rpm, that lead to 3000 rpm of engine speed (2EE0*0.25=3000Dec)
If header is on, you will see response:

```
80 BE E0 2E

>
```

where 80 is MID. If the data bytes from response of PID is longer than 19, it will continuously display long data bytes after first PID byte.

We know PID =00 is a request PID. So if you type 00BE, it means you will request PID BE. It's the same as typing BE. However, you can request more PIDs in one line command. For example, you want engine speed and fuel temperature (PID=174 decimals or AE hexadecimal), you can type

```
>00 BE 00 AE
BE E0 2E AE 40 01

>
```

The response is BE E0 2E AE 40 01. The first byte is PID, So the engine speed will be 2EE0, and AE after 2E is PID of fuel temperature, fuel temperature is 0140, that is 320 decimals. Actual temperature is 320x0.25=80 Fahrenheit degree.

Note: *Response sometimes does not follow the PID request order, for the above example it may respond to "AE 40 01 BE E0 2E"*

Sometimes, we maybe get "No Data" response, we can increase command time out value by AT ST hh command. If DFL168A didn't return ">" for long time because of waiting for truck response, you can use "Ctrl+C" to pause the waiting and return ">", you can execute some non-truck data buss access command. Afterwards, DFL168A can return data response from truck when data is ready.

DFL168A can enter monitor mode by typing command "AT MA" or "AT MMID 00 " or "MMID xx" or "MMID xx 00" or "MMID xx hh ... qq" in hyperterminal.

DFL168A will continuously display the MID and PID which it monitored. Pressing any key in the hyperterminal will exit monitor mode and return ">" normal state. Display in the monitor mode is the same as one of normal state. However, "AT MA" command of J1708/J1587 will not combine multisection PID to a over-sized PID.

DFL168A can send non-request PID, only you keep that the data bytes are less than 19 bytes. DFL168A will automatically add MID and check sum for you. Like J1939, in this way you can make a simple J1708/J1587 simulator.

Note: *You have to fully understand the J1708/J1587 protocol if you want to send non-request PID, otherwise you may damage the vehicle.*

Furthermore, like J1939, J1708/J1587 broadcast most of messages periodically. In most of time, we don't need actually send a request to truck data bus. The command "AT INTRUDE 0" will make DFL168A not to send any message to truck data bus. But DFL168A can still get response if there is a broadcast for the request PID. This way makes DFL168A no any intruder to truck data bus.

Vehicle speed pulse output is spontaneous, you don't need to send speed request (PID 54 Hex) because PID 54 Hex broadcasts every 0.1Seconds . But vehicle forward/backward output depends on your PID 64 request. When vehicle speed is very low, you have to send PID 40 Hex request to modify the vehicle forward/backward output. You only need send PID 40 Hex request, you don't need write output port, DFL168A automatically do it for you.

6.4.3 ISO15765

The ISO15765 standard is an OBD2 protocols used in middle or light duty vehicle. It uses ISO 11898 standard CAN physical interface, and defines its data format and transfer.

DFL168A supports 4 kind of ISO15765

1. CAN ID has 11 bits and baudrate is 500Kbps. ISO15765 Protocol number is 6, you can use "AT SP 6" (stored your selected protocol) or "AT TP 6" (Not stored your selected protocol) to select it
2. CAN ID has 29 bits and baudrate is 500Kbps. ISO15765 Protocol number is 7, you can use "AT SP 7" (stored your selected protocol) or "AT TP 7" (Not stored your selected protocol) to select it
3. CAN ID has 11 bits and baudrate is 250Kbps. ISO15765 Protocol number is 8, you can use "AT SP 8" (stored your selected protocol) or "AT TP 8" (Not stored your selected protocol) to select it.
4. CAN ID has 11 bits and baudrate is 250Kbps. ISO15765 Protocol number is 9, you can use "AT SP 9" (stored your selected protocol) or "AT TP 9" (Not stored your selected protocol) to select it

Of cause, if you don't know which ISO15765, you can use "AT SP 0" or "AT TP 0" to set correct ISO15765 automatically. However, Protocol number 0 with searching automatically can be only used ISO15765, you cannot use it in J1939, J1708 and User1/User2 protocols.

If baud rate is not 500K and 250K, you can choose User1 and User2 protocol by setting correct parameters.

DFL168A provide the functions below for ISO15765 Protocol:

1. Support mixed addressing mode (You can use AT AD1 to turn on mixed address, default is off)
2. Support multi-frame response automatically.
3. Flow control Data is adjustable.
4. Automatically output vehicle speed pulse, and frequency can be scaled by PP 1D

How to use OBD? Let us review OBD Standard.

The OBD Standards specify that each request that is sent to the vehicle must adhere to a set format. The first byte sent is known as "Service #" which describes the type of data being requested. And the second byte (probably a third or more) is known as PID (Parameter identification) which specifies actual parameter we require, for example, engine rpm's PID is hex 0C. The details about Service # and PID are in the SAE J1979 or ISO 15031-5 standards document.

SAE Standard J1979 defined 9 diagnostic test services, which are shown below:

- 01------Request Current Powertrain Diagnostic Data
- 02------Request Powertrain Freeze Frame Data
- 03------Request Emission-Related Diagnostic Trouble Codes
- 04------Clear/Reset Emission-Related Diagnostic Information
- 05------Request Oxygen Sensor Monitoring Test Results
- 06------Request On-Board Monitoring Test Results for Specific Monitored Systems
- 07------Request Emission-Related Diagnostic Trouble Codes Detected
- 08------Request Control of On-Board System, Test or Component
- 09------Request Vehicle Information

Vehicle is not required to support all services, and all PIDs. However, for every service #, PID=00 must be supported.The value of PID 00 is 32 bits unsigned integer that tells us which PID will be supported from PID 01 to PID 32. It is bit-encoded. Most Significant Bit (MSb) denotes PID 01, LSb denotes PID 32. If value of bit is 1, it means the PID will be supported, and 0 means the PID will not be supported.

If PID 32 (hex is 0x20) is supported, the value of PID 32 is 32 bits unsigned integer that tells us which PID will be supported from PID 33 to PID 64. Most Significant Bit (MSb) denotes PID 33, LSb denotes PID 64. The same rule is for PID 64 (0x40), PID 96 (0x60), ...

For our DFL168A, at the hint prompt, issue the service # 01 PID 00 command:

```
> 01 00
```

the response could be as follows:

```
41 00 BE 1F B8 10
```

The 41 denotes a successful response from service # 1 request (0x40+ 0x01=0x41) . If service # 02 request, the first byte of response is 0x42, service # 03 with 0x 43, service # 04 with 0x 44, ..., etc. The second byte 0x00 just repeats PID 00 we request. The next 4 bytes (BE 1F B8 10) denotes the value of requested PID 00.

Another example is for reading coolant temperature which is PID 05 in service #01.

So type command after hint prompt:

```
>01 05
```

We get response

41 05 7C

41 05 tell us it is successful response from Service #01 and PID 05. Hex 7C denotes value of PID 05. It is decimal 124 which represents 84 Celsius degree because 0 denotes -40 Celsius degree (124-40=84)

The example above is single frame response. If response data (including service # and PID) is more than 7 bytes, it will display in multiple frames. For example, we request VIN#. We will use service # 09 and PID 02. So type command after hint prompt:

>09 02

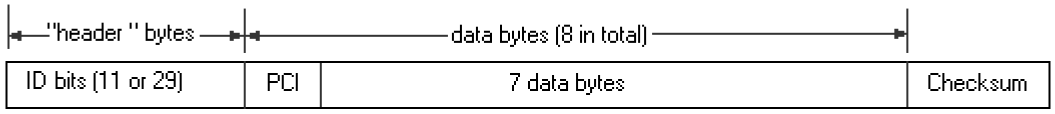
We get response

```
014
0: 49 02 01 44 41 46
1: 55 4C 41 49 45 4C 45
2: 43 54 52 4F 4E 49 43
```

The first line 014 tells us total 20 (hex 014 = decimal 20) bytes response. The second line 0 before colon (:) is frame number, and the data after colon (:) is actual response data. In the same way, the third line 1 before colon (:) is frame number, and the data after colon (:) is actual response data. The 4th line 2 before colon (:) is frame number, and the data after colon (:) is actual response data. So the actual response data will be "49 02 01 44 41 46 55 4C 41 49 45 4C 45 43 54 52 4F 4E 49 43" which has 20 bytes. If response data length is more than 20, we will get rid of the right side remainder bytes.

Let's see the actual respond data. "49 02" denotes service # 09 and PID 02, the 3rd byte "01" denotes "Number of data items" which is fixed 01 in J1979. The remainder data will be ASCII code of VIN# which is "DAFULAIELECTRONIC"

The ISO 15765-4 protocols uses the flowing structure to encapsulate OBD2 data.



For checksum, CAN Bus takes care of it automatically, you just ignore it.

In general, you only can see 7 data bytes, you don't need to see ID bits and PCI in you UART except you turn on header display by "AT H1" command.

When you turn on header display, you can see ID bits, and PCI .

If you send request or other data to Can bus via UART, you don't need set header and PCI. DFL168A adds these bytes for you automatically. However, if you want to change the default header , you will run the following "AT Command":

1. For 11 bits ISO15765, you can use "AT SH hhh" to set up 11 bits header, and it only takes the right-most 11 bits of hex hhh.
2. For 29 bits ISO15765, you can use "AT SH hh hh hh" to set up the right-most 24 bits of 29 bits

header, and use "AT CP hh " to set up the left-most 5 bits of 29 bits header.

If you want to send PCI by yourself, you can execute "AT CF0" Command. But you must cancel flow control by "AT CFC0" Command, otherwise, DFL168A still add PCI for you automatically. For example, we run command below:

```
>AT CAF 0
OK
>AT CFC 0
OK
```

Now you should prepare PCI by yourself. We want serice#01 and PID 00. In the past, we only need to type 01 00. But now you should prepare PCI. The PCI should be hex 02. The "0" denotes single frame, The "2" denotes 2 data bytes (for 01 00). So you should type command below:

```
>02 01 00
```

You will get response below:

```
06 41 00 BE 1F B8 10 00
>
```

First byte is response PCI which is 06. The "0" denotes single response frame. The "6" denotes 6 data response bytes which are "41 00 BE 1F B8 10".

If you want to resume automatic PCI, you should execute "AT CAF 1".

For normal addressing, the maximum data response length is 7 for single frame. And for mixed addressing, the maximum data response length is 6 for single frame. If response data is more than 7 (6), it will use multi-frame response. DFL168A must send flow control in this situation. Fortunately, DFL168A automatically send flow control, you need do nothing. This is default Flow control mode 0. Sometimes, you want some flexible for flow control. So we provide 3 Flow control mode:

1. Flow control mode 0: All Flow controls including header and data are provided by "DFL168A" automatically. Usually you use this mode, and it's default mode.
2. Flow control mode 1: Flow control header is decided by "AT FC SH hhh" (11 bits ID) or "AT FC SH hh hh hh hh" (29 bits ID). And Flow control data are decided by "AT FC SD [1-5 bytes]" command. Of cause, Flow control will be sent out automatically when it is needed.
3. Flow control mode 2: Flow control header is decided by DFL168A automatically. But Flow control data are decided by "AT FC SD [1-5 bytes]" command. Of cause, Flow control will be sent out automatically when it is needed.

After you set up Flow control header and data, you can use "AT FC SM h" to select. h=0,1,2.

Sometimes, You don't need Flow control, for example, you don't accept multiple frame data or you use your data as your flow control just like request command. You can use "AT CFC 0" to turn off Flow control. Of cause, you can use "AT CFC 1" to resume Flow control.

ISO15765 is an request/response protocols. Tester (DFL168A) sends request, and the vehicle responds to the request. We descript header above for the request. For response, our IC DFL168A should set up filters/mask for getting our interesting respond data. Usually, DFL168A set up correct filter/mask automatically for us. You just do nothing, and just get respond data. However, you can set up different Filters/masks by the following "AT commands"

1. "AT CF hhh" command will set up 11 bits ID Filter for 11 bits ISO15765 Protocol, and "AT CM hhh" will set up 11 bits ID Mask for 11 bits CAN bus Protocol. Only data with special ID will be accepted by DFL168A. The special ID operating Logic "And" with Mask will be equal to Filter operating Logic "And" with Mask.
2. "AT CF hh hh hh hh" command will set up 29 bits ID Filter for 29 bits CAN Bus Protocol , and "AT CM hh hh hh hh" will set up 29 bits ID Mask for 29 bits CAN Bus Protocol. Only data with special ID will be accepted by DFL168A. The special ID operating Logic "And" with Mask will be equal to Filter operating Logic "And" with Mask.
3. "AT CRA hhh" command will set up 11 bits ID Filter and mask for 11 bits CAN Bus Protocol at one command. It tells DFL168A that it only accepts data with ID hhh (Only right-most 11 bits)
4. "AT CRA hh hh hh hh" command will set up 29 bits ID Filter and mask for 29 bits CAN Bus Protocol at one command. It tells DFL168A that it only accepts data with ID "hh hh hh hh" (Only right-most 29 bits)

Above commands cancel DFL168A automatic setting for ID of received data packet. However, if you execute "AT AR" command, DFL168A will resume the automatic setting.

Note: *If you change the Filter/mask of response, you should make sure that it can receive Service # 1 and PID 0D (vehicle speed), otherwise, the pin 26 of DFL168A (VSPEED Pin) will not output speed PWM.*

6.4.4 USER1/USER2

User1 and User2 are fully customized CANbus protocols.

User1's protocol number is hex B, and User2's protocol number is hex C. You can use "AT SP h" or "AT TP h" command to choose it.

User1 and User2 are a request/response protocol. You firstly should set up CAN bus ID (Header) for request before you send request data. Of cause, you will only receive your interesting data response, so you should set up Filter/mask too before you send request data.

User2 protocol is the same as User1. Their behaviour is controlled by "Programmable Parameters" .

The address 0x34 in the "Programmable Parameters" decides the baud rate of User1 protocol. The actual baud rate is 500K/Parameter value. The default value is 0x04 which has 125K baud rate

(500K/4=125Kbps).

The address 0x36 in the "Programmable Parameters" decides the baud rate of User2 protocol. The actual baud rate is 500K/Parameter value. The default value is 0x0A which has 50K baud rate (500K/10=50Kbps).

You can use commands to change USER1 baud rate to 250Kbps as follows below:

```
> AT PP 34 SV 02
OK
>AT PP 34 ON
OK
>AT PPP
```

and you should re-power on DFL168A in order to use new baud rate.

The address 0x33 in the "Programmable Parameters" decides User1 work mode.

It is 8 bits byte. We use "B7 B6 B5 B4 B3 B2 B1 B0" to denote the byte. B7 is left-most bit, B0 is right-most bit.

B7 is for setting the length of CAN BUS ID transmitted. 0: 29 bits ID. 1: 11 bits ID.

B6 is for setting the length of CAN BUS Data transmitted. 0: fixed 8 bytes length. 1: DLC(Data length) is not fixed.

B5 is for setting the length of CAN BUS ID received. 0: Length defined by B7. 1: Both 11 and 29 bits ID can be accepted.

B4 and B3 are reserved.

B2 B1 B0 are for data format. B2B1B0="000" means none format. B2B1B0="001" means ISO15765 format. B2B1B0="010" means SAE J1939 format

The default value of "Programmable Parameters" at 0x33 is 0xE0.

The address 0x35 in the "Programmable Parameters" decides User2 work mode.

Its definition is the same as "Programmable Parameters" at 0x33. The default value of "Programmable Parameters" at 0x35 is 0x80.

Please use "AT SH hhh" to set up 11 bits ID transmitted. And use "AT SH hh hh hh" and "AT CP hh" to set up 29 bits ID transmitted.

For data received, you can set up different Filters/masks by the following "AT commands"

1. "AT CF hhh" command will set up 11 bits ID Filter for 11 bits user Protocol, and "AT CM hhh" will set up 11 bits ID Mask for 11 bits user Protocol. Only data with special ID will be accepted by DFL168A. The special ID operating Logic "And" with Mask will be equal to Filter operating Logic "And" with Mask.
2. "AT CF hh hh hh hh" command will set up 29 bits ID Filter for 29 bits user Protocol , and "AT CM hh hh hh hh" will set up 29 bits ID Mask for 29 bits user Protocol. Only data with special ID will be accepted by DFL168A. The special ID operating Logic "And" with Mask will be equal to Filter operating Logic "And" with Mask.
3. "AT CRA hhh" command will set up 11 bits ID Filter and mask for 11 bits user Protocol at

one command. It tells DFL168A that it only accepts data with ID hhh (Only right-most 11 bits)

4. "AT CRA hh hh hh hh" command will set up 29 bits ID Filter and mask for 29 bits user Protocol at one command. It tells DFL168A that it only accepts data with ID "hh hh hh hh" (Only right-most 29 bits)

And user protocol can send remote frame (No any data) by "AT RTR" command if it is in non-format. Firstly you set the header to the value you want to receive, and then you send "AT RTR" command, and you will receive data with this header. Of cause, you must set filter/mask to this header before you send "AT RTR" command.

When we use User1 and User2, and data transmitted is fixed 8 bytes, however, if we type digital command is less than 8 bytes, DFL168A will automatically use stuff byte to make up all transmitted data to 8 bytes. The stuff byte content is decided by "Programmable Parameters" at address 0x31. the default value is 0x00. You can use "AT PP 31 SV hh" , "AT PP 31 ON" and "AT PPP" command to change to hex hh.

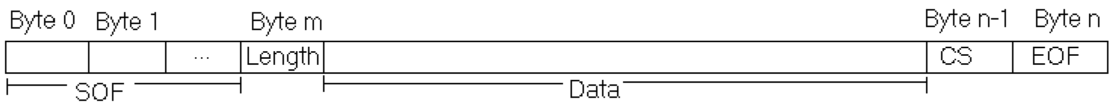
6.5 Device 1 Access

We use UART2 as Device 1 access. UART2 baud rate ranges from 2400bps to 5000/3 kbps. It can be changed by PP 37 and UART2 has 1 Start bit, 8 data bits and 1 Stop bit. DFL168A supports UART2 which connects most of RS232 Devices. Current version supports 7 devices. If your device does not occur in the list, please use "General Binary Device" or "General ASCII Device". They are flexible device which can be configured by "AT DEV1 " command. If "General Binary Device" or "General ASCII Device" cannot work with your device, please use "Transparent mode" which means that all transmit and receive of UART2 will be changed to "UART1". It looks like external device connects UART1 directly.

6.5.1 General Device with Binary response

"General Device with binary response" is a device which use binary for data bytes.

The data packet format is below:



The data field will be binary. SOF (Start of frame) or header will be at most 6 byte. And it can be 0 byte. Length will be one byte or 0 byte. CS can be one byte or 2 bytes or 0 byte, EOF (End of Frame) or Tail can be one byte or 0 byte. Data can be one byte to 245 bytes.

SOF can has at most 5 different header for one device, we call it SOF number, which is from 1 to 5.
EOF must be the same for one device.
CS must be the same for one device.

When quantity of SOF bytes is 0, quantity of length byte is 0, quantity of CS byte is 0, and quantity of EOF byte is 0, Data packet is degraded into only data field. We only depend on the time separation to separate different data packet. This will be easily cause data packet lost. We should avoid this situation.

The commands "AT DEV1 SOF h" and "AT DEV1 SOF h xx ... mm" will set up SOF of data packet, please read 6.9 AT Command Details: DEV1 SOF h

The commands "AT DEV1 EOF" and "AT DEV1 EOF hh" will set up EOF of data packet, please read 6.9 AT Command Details: DEV1 EOF

DFL168A has 7 check sum modes. It is set by "AT DEV1 CSM hh" command, please read 6.9 AT Command Details: DEV1 CSM hh

If your device contain check sum which is different from our definition, you can choose no check sum, and you will get all data including check sum, and then you can verify the receiving data using receiving check sum by yourself.

EOF can exist, and may not exist. You use "AT DEV1 EOF" to cancel EOF, and use "AT DEV1 EOF hh" to set an EOF to hh (Hex)

Our definition of data packet length is all bytes including SOF and Length and CS and EOF.
If receiving data packet has a fixed length, length byte in one data packet will not be necessary. You can use "AT DEV1 LEN hh" to set up the fixed length to hh (Hex). You can use "AT DEV1 LEN" to cancel length, it means that data packet has no any length, it's only identified by EOF or time between bytes.
When receiving data packet has length, but length is not fixed, you can use length byte to get length information. However, some devices have different definition of length from us. Its length may not contains SOF or or CS or EOF. So we must have a method to translate length of customer device into our definition. The command "AT DEV1 LEN xx hh " will do this job. Please read 6.9 AT Command Details: DEV1 LEN xx hh carefully.

It is allowable that space time (or called idle time) between bytes within one data packet exists. However if the space is too long, DFL168A will think the data packet is broken. And DFL168A will throw any the data packet. The command "AT DEV1 BT hh" will set the broken time hh x 4 ms. The space time definition is time from previous byte stop bit to current byte stop bit.

The receiving data packet can be identified by space time (or called idle time). If space time is longer enough, DFL168A think a data packet ends. The commands "AT DEV1 S hh" and "AT DEV1 S" will set or cancel space between data packets. Please read 6.9 AT Command Details: DEV1 S carefully

If data packet is identified only by separating time between packets (no length, no SOF , no EOF), broken time which is set by "AT DEV1 BT hh" must bigger than separating time which is set by "AT DEV1 S hh", otherwise, you will get "No data"

We should use SOF and EOF and length information of dada packet as possible as we can, it will help us identify data packet exactly.

When header is on (AT H1), you will see all receiving data bytes except check sum and EOF.
When header is off (AT H0), you will only see bytes of data field.

We open devices by command "AT DEV1 SP h" or "AT DEV1 TP h". The default device 1 is "General Device with binary response" except you once used "AT DEV1 SP h" in the latest operation of DFL168A.
DFL168A will automatically open the default device1when power on even though you have no executed command "AT DEV1 SP h" or "AT DEV1 TP h". It will automatically receive response from UART2 and put them into history buffers.

The command "AT DEV1 GB" will get the broadcast data from Device 1 no matter which SOF it broadcasted.
The command "AT DEV1 GB SOF h " will get the broadcast data with SOF number h from Device 1
The command "AT DEV1 REQ hh hh ... hh" (Quantity of hex must be even) will send binary message " hh hh ... hh" to Device 1, and listen the response from Device 1. Which SOF will it listen? You have to set it by command "AT DEV1 RSOF h" before you send command "AT DEV1 REQ hh hh ... hh". In command "AT DEV1 RSOF h", h is SOF number you want to listen. h=0 denotes we don't care SOF, DFL168A will receive any SOF.

Note: *In command "AT DEV1 REQ hh ... hh" , you must send the header and CS and Tail by yourself if they exist. DFL168A didn't do it for you. (in "hh...hh" , type your header or tail or CS hex)*

When you select "General Device with binary response" , the following defaults have been set by DFL168A automatically:

(We use C++ style to comment each AT command even though DFL168A does not support comments)

```
AT DEV1 S 07          //Data Separate time is 7x4=28ms
AT DEV1 ST 7D         // Maximum of waiting time is 7Dx4=500ms for Device 1 request response
AT DEV1 BT 0A         // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN           // No Length Data Packet
AT DEV1 SOF 1         //No SOF number 1
AT DEV1 SOF 2         //No SOF number 2
AT DEV1 SOF 3         //No SOF number 3
AT DEV1 SOF 4         //No SOF number 4
AT DEV1 SOF 5         //No SOF number 5
AT DEV1 EOF           //No EOF
AT DEV1 HT 0          // Don't use history buffer for request response
AT DEV1 HFC 0         //No Hardware Flow control RTS2/CTS2
AT DEV1 CSM 00        //No Check Sum
AT DEV1 BRD 00        // Baud rate is 9600bps
```

The default baud rate is 9600bps. You can temporally change the baud rate by command "DEV1 BRD hh ". If you want permanent change, please use PP 37. Other default values can be changed by AT commands.

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

6.5.2 General Device with ASCII response

"General Device with ASCII response" is a device which use ASCII for data bytes.

It is almost the same as "General Device with Binary response" except data field is ASCII characters.

Most of AT command are the same as "Binary Device"

The command "AT DEV1 REQ hh hh ... hh" is different. The quantity of "hh ... hh" may be odd. It is a ASCII characters. For space or non-display characters, you have to use escape character "\"

Escape character	Hexadecimal value
\0	00
\n	0A
\r	0D
\t	09
\v	0B
\a	07
\f	0C
\'	27
\"	22
\?	3F
\\	5C
\xhh	hh

Like "General Device with Binary response", you are responsible for any SOF and Length and CS and EOF of sending data packet. DFL168A didn't do it for you. However, SOF and Length and CS and EOF must be binary, you have to use "\xhh" to send the binary data.

Display of response for "General Device with ASCII response" is different, Data field will be displayed in ASCII characters. However, SOF and Length will be displayed in Hex.

When you select "General Device with ASCII response" , the following defaults have been set by DFL168A automatically:
(We use C++ style to comment each AT command even though DFL168A does not support comments)

```
AT DEV1 S 07          //Data Separate time is 7x4=28ms
AT DEV1 ST 7D         // Maximum of waiting time is 7Dx4=500ms for Device 1 request response
AT DEV1 BT 0A         // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN           // No Length Data Packet
AT DEV1 SOF 1         //No SOF number 1
AT DEV1 SOF 2         //No SOF number 2
AT DEV1 SOF 3         //No SOF number 3
AT DEV1 SOF 4         //No SOF number 4
AT DEV1 SOF 5         //No SOF number 5
AT DEV1 EOF           //No EOF
AT DEV1 HT 0          // Don't use history buffer for request response
AT DEV1 HFC 0           //No Hardware Flow control RTS2/CTS2
AT DEV1 CSM 00          //No Check Sum
AT DEV1 BRD 00          // Baud rate is 9600bps
```

The default baud rate is 9600bps. You can temporally change the baud rate by command "DEV1 BRD hh ". If you want permanent change, please use PP 37. Other default values can be changed by AT commands.

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

You can configure the device1 as ASCII Modbus, but right now we didn't provide CRC verification. You have to set no check sum, and you do the CRC verification by yourself.

6.5.3 CS440 controller

Bosch Rexroth CS-440 controller can be connected to UART2.
Actually, it belongs to a special "General Device with binary response" with special setting. All commands for "General Device with binary response" can be used in the "CS440 controller".

When you select "CS440 controller" , the following defaults have been set by DFL168A automatically:
(We use C++ style to comment each AT command even though DFL168A does not support comments)

```
AT DEV1 S               //No Data Packet Separate
AT DEV1 ST FC           //Maximum of waiting time is 4Seconds for Device 1 request response
AT DEV1 BT 0A           // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN 58          // 88 bytes' Fixed Length of Data Packet
AT DEV1 SOF 1           //No SOF number 1
AT DEV1 SOF 2           //No SOF number 2
AT DEV1 SOF 3           //No SOF number 3
AT DEV1 SOF 4           //No SOF number 4
AT DEV1 SOF 5           //No SOF number 5
AT DEV1 EOF             //No EOF
```

AT DEV1 HT 0 // Don't use history buffer for request response
AT DEV1 HFC 0 //No Hardware Flow control RTS2/CTS2
AT DEV1 CSM 82 //Double byte Check Sum, First LSB for Checksum word, Check sum mode is 2
AT DEV1 CS 0000 //extra Checksum is 0
AT DEV1 BRD 2B // Baud rate is 115200bps

.

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

There is a special commands "AT DEV1 REQ" for CS440. It will wait for 86 bytes' response from CS440. Version1.00 didn't provide parse command for "CS440". You have to parse the 86 bytes' data by yourself. In future, we will provide the parse command , you only need to send "AT DEV1 SI REQ" and then use the parse commands to get parameters.

6.5.4 CS230 Controller

Bosch Rexroth CS230 controller can be connected to UART2.
Actually, it belongs to a special "General Device with binary response" with special setting. All commands for "General Device with binary response" can be used in the "CS230 controller".

When you select "CS230 controller" , the following defaults have been set by DFL168A automatically:
(We use C++ style to comment each AT command even though DFL168A does not support comments)

AT DEV1 S //No Data Packet Separate
AT DEV1 ST 7D // Maximum of waiting time is 7Dx4=500ms for Device 1 request response
AT DEV1 BT 0A // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN 02 00 // Length byte is in the position 2 of receive byte, (Position 0 is first byte)
AT DEV1 SOF 1 //No SOF number 1
AT DEV1 SOF 2 //No SOF number 2
AT DEV1 SOF 3 //No SOF number 3
AT DEV1 SOF 4 //No SOF number 4
AT DEV1 SOF 5 //No SOF number 5
AT DEV1 EOF //No EOF

```
AT DEV1 HT 0           // Don't use history buffer for request response
AT DEV1 HFC 0          //No Hardware Flow control RTS2/CTS2
AT DEV1 CSM 02         //Single byte Check Sum, Check sum mode is 2
AT DEV1 CS 00          //extra Checksum is 0
AT DEV1 BRD 01         // Baud rate is 4800bps
```

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

CA230 uses "map address" to express different parameters (like PID), map address is 16 bit integer, and parameters can be single byte, or double bytes or 4 bytes. We provide a special command "AT DEV1 ADR REQ hhhh" for request parameter of map address "hhhh". The relation of map address and its parameter will be in the Manual of "CS-230 Communication Protocol for GPS Application"

6.5.5 Dickey-John Controller

DICKEY-John controller can be connected to UART2.

Actually, it is "General Device with ASCII response" with special setting. All commands for "General Device with ASCII response" can be used in the "DICKEY-John controller"

When you select "DICKEY-John controller" , the following defaults have been set by DFL168A automatically:
(We use C++ style to comment each AT command even though DFL168A does not support comments)

```
AT DEV1 S              //No Data Packet Separate
AT DEV1 ST 7D          // Maximum of waiting time is 7Dx4=500ms for Device 1 request response
AT DEV1 BT 0A          // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN 02 04          // Length byte is in the position 2 of receive byte, (Position 0 is first byte), Length byte content
                       //  will be added 04
AT DEV1 SOF 1 24 F0    // SOF number 1 is $F0
AT DEV1 SOF 2 24 F1    // SOF number 2 is $F1
AT DEV1 SOF 3 24 F2       // SOF number 3 is $F2
AT DEV1 SOF 4 24 F3       // SOF number 4 is $F3
AT DEV1 SOF 5 24 FB      // SOF number 5 is $FB

AT DEV1 EOF               //No EOF
AT DEV1 HT 0              // Don't use history buffer for request response
AT DEV1 HFC 1             // Hardware Flow control RTS2/CTS2 is enabled
AT DEV1 CSM 01           //Single byte Check Sum, Check sum mode is 01
AT DEV1 CS 55             //extra Checksum is 55
AT DEV1 BRD FF            // Baud rate is 19200bps
```

.

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

DICKEY-John controller needs to be initialized before UART2 communication with it. DFL168A did the initialization for you. When DFL168A start DICKEY-John controller because power on or running "AT SP/TP h" command, DFL168A will send initializing sequence to DICKEY-John controller automatically.

DICKEY-John controller uses "DATA_GROUP" "DATA_ELEMENT" to express different parameters (like PID). "DATA_GROUP" is 8 bits integer, "DATA_ELEMENT" is 8 bits integer too. Sometimes, DICKEY-John controller uses "DATA_SUB_ELEMENT" too. "DATA_SUB_ELEMENT" is 8 bits integer too. In this situation, 3 items ("DATA_GROUP", "DATA_ELEMENT","DATA_SUB_ELEMENT") will identify a parameter.

DFL168A provides special commands for parameter request. The command "AT DEV1 DJ RD xx yy" will request parameter with DATA_GROUP being xx(Hex) and DATA_ELEMENT being yy(Hex). The command "AT DEV1 DJ RD xx yy zz" will request parameter with DATA_GROUP being xx(Hex) and DATA_ELEMENT being yy(Hex) and DATA_SUB_ELEMENT being zz(Hex).

DICKEY-John controller data field values are the following items:

- Binary Number Representation
  The only binary representation defined is for 8 bit unsigned integer numbers.
- ASCII Number Strings
  All representation of numbers in ASCII strings is in base 10 and either integer, fixed point, or floating point. All ASCII strings are NULL (0) terminated.

So, when header is off, display of response is that the first byte is hexadecimal DATA_ELEMENT, and then it displays parameter value. If parameter value is 8 bit unsigned integer number, it will display "H" character after the 8 bit unsigned integer number (Hex). All ASCII Number Strings will be displayed with NULL (0) terminated.

When header is on, display of response will be that the first byte is hexadecimal 24 (that is '$') and then it is hexadecimal DATA_GROUP and then 1 byte's hexadecimal data packet length and then the contents which are the same as ones when the header is off.

We strongly recommend you using commands "AT DEV1 DJ RD xx yy" and "AT DEV1 DJ RD xx yy zz" instead of "DEV1 REQ hh hh ... hh" . The reasons have two. One reason is the display format will be different from the above. The other reason is that you must prepare the request data by yourself. If any mistakes happen, it maybe damage DICKEY-John controller.

6.5.6 ACE Spreader Controller

ACE, Accent Electronic Controls Inc "ACE Spreader Controller" can be connected to UART2.

Actually, it belongs to a special "General Device with binary response" with special setting. All commands for "General Device with binary response" can be used in the "ACE Spreader".

When you select "ACE Spreader" , the following defaults have been set by DFL168A automatically:
(We use C++ style to comment each AT command even though DFL168A does not support comments)

```
AT DEV1 S            //No Data Packet Separate
AT DEV1 ST 7D          //Maximum of waiting time is 7Dx4=500ms for Device 1 request response
AT DEV1 BT 0A          // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN 15         // 21 bytes' Fixed Length of Data Packet
AT DEV1 SOF 1 11       // SOF number 1 is 11
AT DEV1 SOF 2 12       // SOF number 2 is 12
AT DEV1 SOF 3 13       // SOF number 3 is 13
AT DEV1 SOF 4          //No SOF number 4
AT DEV1 SOF 5          //No SOF number 5
AT DEV1 EOF            //No EOF
AT DEV1 HT 1           // Use history buffer for request response
AT DEV1 HFC 0          //No Hardware Flow control RTS2/CTS2
AT DEV1 CSM 0          //No Check Sum
AT DEV1 BRD FF         // Baud rate is 19200bps
```

.

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

"ACE Spreader Controller" is a broadcast protocol. It will broadcast 3 different data packets, called AC1, AC2 and AC3. And it will broadcast when it thinks it's necessary (not periodically). So if you miss a broadcast, you will lose some message. Therefore, we use history buffer to store these broadcasts.

There are some special commands below for "ACE Spreader":

1. DEV1 AC1

   It will get AC1 broadcast. AC1 contains date time and controller ID and Operator ID, it will broadcast once when controller power on or when operator ID was changed. So when you send "AT DEV1 AC1", DFL168A looks for the broadcast from UART2 firstly. It will use the history data if it does not exist.

2. DEV1 AC2

   It will get AC2 broadcast. AC2 contains date time and information of controller setting, it will broadcast once when controller power on or when operator changed the setting. So when you send "AT DEV1 AC2", DFL168A looks for the broadcast from UART2 firstly. It will use the history data if it does not exist.

3. DEV1 AC3

   It will get AC3 broadcast. AC3 contains date time and information from the actual performance of the spreading process, it will broadcast when controller thought it's necessary. So when you send "AT DEV1 AC3", DFL168A looks for the broadcast from UART2 firstly. It will use the history data if it does not exist.

4. DEV1 ACE DIST

   This command will get the total distance traveled which contains in AC3 broadcast. The parameter in AC3 is a value from previous broadcast to current broadcast. DFL168A will automatically accumulate the distance traveled even though you didn't send command "AT DEV1 AC3" or "DEV1 ACE DIST". So you don't worry missing AC3 broadcast.

5. DEV1 ACE DIST hhhh

   This command will set up the initial value of the distance traveled to hhhh (Hex). You must set the initial value because DFL168A is only responsible for accumulation. "DEV1 ACE DIST 0000" will clear the distance traveled to zero.

6. DEV1 ACE QtyOfS

   This command will get the total quantity of solid material spread which contains in AC3 broadcast. The parameter in AC3 is a value from previous broadcast to current broadcast. DFL168A will automatically accumulate the quantity of solid material spread even though you didn't send command "AT DEV1 AC3" or "DEV1 ACE QtyOfS". So you don't worry missing AC3 broadcast.

7. DEV1 ACE QtyOfS hhhh

   This command will set up the initial value of the quantity of solid material spread to hhhh (Hex). You must set the initial value because DFL168A is only responsible for accumulation. "DEV1 ACE QtyOfS 0000" will clear the quantity of solid material spread to zero.

8. DEV1 ACE QtyOfL

   This command will get the total quantity of liquid spread which contains in AC3 broadcast. The parameter in AC3 is a value from previous broadcast to current broadcast. DFL168A will automatically accumulate the quantity of liquid spread even though you didn't send command "AT DEV1 AC3" or "DEV1 ACE QtyOfL". So you don't worry missing AC3 broadcast.

9. DEV1 ACE QtyOfL hhhh

   This command will set up the initial value of the quantity of liquid spread to hhhh (Hex). You must set the initial value because DFL168A is only responsible for accumulation. "DEV1 ACE QtyOfL 0000" will clear the quantity of liquid spread to zero.

6.5.7 GPS with NMEA 0183 output

In general, AVL (Automatic Vehicle Locator) contains GPS device. Why does DFL168A provide this function?
The reason we provide this function is that if you connect a wireless modem, and you can use DFL168A to make an AVL.

Actually, UART2 belongs to a special "General Device with ASCII response" with special setting. All commands for "General Device with ASCII response" can be used in the "GPS Device".

When you select "GPS with NMEA 0183 output" , the following defaults have been set by DFL168A automatically:
(We use C++ style to comment each AT command even though DFL168A doesnot support comments)

```
AT DEV1 S                   //No Data Packet Separate
AT DEV1 ST FB               // Maximum of waiting time is 2 Seconds for Device 1 request response
AT DEV1 BT 0A               // Data Packet broken time is 0Ax4=40ms
AT DEV1 LEN                 // No Length Data Packet
AT DEV1 SOF 1 24 47 50  52 4D 43    // SOF number 1 is $GPRMC
AT DEV1 SOF 2 24 47 50  47 47 41     // SOF number 2 is $GPGGA
AT DEV1 SOF 3 24 47 50  56 54 47     // SOF number 3 is $GPVTG
AT DEV1 SOF 4               //No SOF number 4
AT DEV1 SOF 5               //No SOF number 5
AT DEV1 EOF  0D             // EOF is ' \r '
AT DEV1 HT 0                // Don't use history buffer for request response
AT DEV1 HFC 0               //No Hardware Flow control RTS2/CTS2
AT DEV1 CSM 0               //No Check Sum
AT DEV1 BRD 01              // Baud rate is 4800bps
```

. We use a special "Check Sum" for GPS device even though the above setting is no check sum. This check sum is "exclusive OR of all characters" instead of arithmetical sum.

Note: *You have to use command "AT DEV1 PC" before you change any setting. You use command "AT DEV1 SP h" or "AT DEV1 TP h" to open device after you did all changes.*

The format of display when header is on is different from "General Device with ASCII response". SOF will be displayed in ASCII characters instead of Hex. No matter whether header is on, Check Sum will be displayed.

We will receive 3 SOF data packets, $GPRMC, $GPGGA, and $GPVTG. You will use command "AT DEV1 GB SOF 1" to get $GPRMC sentence, "AT DEV1 GB SOF 2" to get $GPGGA sentence, "AT DEV1 GB SOF 3" to get $GPVTG sentence

There are some special parse GPS NMEA183 sentence commands below:

1. DEV1 Alt

   This command is used for getting the altitude from GPS (Device 1). We know that altitude is available for GGA sentence. So we have to get GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 2
OK

>AT DEV1 ALT
1080

>
```

It means the antenna of GPS is 1080 meters above mean-sea-level.

2. DEV1 Date

   This command is used for getting the date from GPS (Device 1). We know that date is available for RMC sentence. So we have to get RMC broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 DATE
25/01/2012

>
```

   It means 25th day of January 2012

3. DEV1 Lat

   This command is used for getting the latitude from GPS (Device 1). We know that latitude is available for RMC or GGA sentence. So we have to get RMC or GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 LAT
+25 33.456'

>
```

   It means latitude is north 25 degree 33.456 minutes

4. DEV1 Long

   This command is used for getting the longitude from GPS (Device 1). We know that longitude is available for RMC or GGA sentence. So we have to get RMC or GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 2
OK

>AT DEV1 LONG
-105 33.4567'

>
```

It means longitude is west 105 degree 33.4567 minutes

5. DEV1 Speed

   This command is used for getting the speed of vehicle from GPS (Device 1). We know that speed is available for RMC or VTG sentence. So we have to get RMC or VTG broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 3
OK

>AT DEV1 SPEED
56.7

>
```

It means the speed is 56.7 knots

6. DEV1 Time

   This command is used for getting the time from GPS (Device 1). We know that time is available for RMC or GGA sentence. So we have to get RMC or GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 TIME
20:58:31

>
```

Time display format is hh:mm:ss

If a sentence contains your all parameters, you only need send one "AT DEV1 SI GB SOF h" command. For example:

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 TIME
20:58:31
```

```
>AT DEV1 LAT
+25 33.456'

>AT DEV1 LONG
-105 33.4567'
```

6.5.8 Transparent mode

In this transparent mode, all communication between external device and UART2 looks like communication between external device and UART1. It means that UART1 connects with UART2. It is useful for unknown external device.
We use "AT DEV1 Transp" to enter "transparent mode". After executing this command, UART1 cannot access vehicle bus data (J1939/J1708). So you cannot use "transparent mode" for long time except you don't care vehicle bus data.
You can use time out to exit "transparent mode" and enter normal command mode. Time out value is set up by "AT DEV1 ST hh" command.
and you can disable "time out " to exit "transparent mode" by "AT DEV1 TPT 0" command. Of cause , you can use command "AT DEV1 TPT 1" to enable "time out " to exit "transparent mode" again..
After ending "transparent mode", you can access vehicle bus data again.
It has another way to exit "transparent mode" by sending a special character to UART1. This special character can be set by "AT DEV1 Exit T hh", hexadecimal hh is ASCII code of this special character. Default character is "ESC" (Ascii code is 0x1b, that is to say, hh=1b). You must avoid this special character occurring in normal data which transmit to external device, otherwise, normal data transmitting to external device will cause unexpected ending of "transparent mode". In order to avoid this situation, you can use command "AT DEV1 TPK 0" to disable this special character. Of cause, you can use command "AT DEV1 TPK 1" to enable this special character again.

In order to avoid data overflow, we suggest the baud rate of UART1 is equal to or bigger than one of UART2. Of cause, the baud rate of UART1 may be less than the baud rate of UART2 because UART1 has 255 bytes' buffer. However, the burst data from external device to UART2 must be less than 255 bytes in this situation.

In this transparent mode, you can select any Device 1 protocol. Device 1 protocol decide the baud rate of UART2.
Our MCU firmware which communicates with UART1 should be like followings:

```
 Set protocol of vehicle by "AT TP h" command (You don't need if you used "AT SP h" in the
 past)
 Set protocol of  Device 1 by "AT DEV1 TP h" command (You don't need if you used "AT
DEV1 SP h" in the past)
for ( ; ; )
{
   Read J1939 PGNs, and process PGNs
   Read 1-wires, and process iButton
   Read Discrete Inputs, and process inputs
   Enter  "transparent mode" by "AT DEV1 Transp" command
   Process Data from UART2
   Send special Character to exit  "transparent mode" or time out  to exit  "transparent mode"

}
```

Our "transparent mode" gives customers ability to handle any external device.

Note: *The VSPEED pin of DFL168A still outputs correct PWM for vehicle speed even though you use transparent mode because the DFL168A use multiple tasks handle at the same time.*

6.6 Power Management

DFL168A has an excellent power management. There are 3 ways to make DFL168A enter sleep.

Method 1: Sleep-pin triggered Sleep

Sleep pin will trigger the sleep. Ignition wire connects sleep pin (may need transistor or optocoupler circuit). Sleep active logic level can be changed by command "AT SLEEP P h" . When sleep pin is active, DFL168A will send a warning to hyperterminal:

```
Warning: Sleep in 0012Hex seconds
```

After the sleep delay time, if sleep pin is still active, DFL168A will display

```
Sleep
```

and then delay 20 to 21 ms, PD pin will output sleep Logic level, then DFL168A sleeps. Users will have enough time to do something before users shutdown itself using PD pin

If sleep pin become inactive within sleep delay time, DFL168A will cancel sleep and display:

```
Warning Canceled!
>
```

After sleep, if sleep pin becomes inactive, DFL168A will wake up, and PD pin will output normal logic level and delay 4 to 5 ms and display:

```
IC wakes up
>
```

IC will restore state which is one of before sleep. If IC was monitor mode before sleep, IC will enter normal mode after wake up.

For this sleep method, DFL168A only has as low as 28uA. However, customer should select low power transceiver IC. And user should put transceiver IC into low power standby mode when DFL168A sleep, such as CAN BUS transceiver MCP2561 or MCP2562 STBY pin connecting with DFL168A pin PD. Otherwise, total current will be increased.

Method 2: Vehicle data bus non-activity triggered Sleep

If DFL168A has not receive any signal from vehicle data bus for 5 seconds, it will trigger sleep warning. DFL168A will send a warning to hyperterminal:

```
Warning: Sleep in 0012Hex seconds
```

After the sleep delay time, if no any signals from vehicle data bus are received by IC, DFL168A will display

```
Sleep
```

and then delay 20 to 21 ms, PD pin will output sleep Logic level, then DFL168A sleeps. Users will have enough time to do something before users shutdown itself using PD pin.

If some bytes from vehicle data bus are received by IC within sleep delay time, DFL168A will cancel sleep and display:

```
Warning  Canceled!
>
```

After sleep, if any signals from vehicle data bus occur, DFL168A will wake up ,and PD pin will output normal Logic level and delay 4 to 5 ms and display:

```
IC wakes up
>
```

IC will restore state which is one of before sleep. If IC was monitor mode before sleep, IC will enter normal mode after wake up.
For this sleep method, DFL168A only has as low as 28uA. However, customer should select low power transceiver IC. And user should put transceiver IC into low power standby mode when DFL168A sleep, such as CAN BUS transceiver MCP2561 or MCP2562 STBY pin connecting with DFL168A pin PD. Otherwise, total current will be increased.

Method 3: SoftwareCommand triggered Sleep

In this way, you send command "AT PC",it will trigger sleep warning. DFL168A will send a warning to hyperterminal:

```
Warning:  Sleep  in 0012Hex seconds
```

This way is the same as method 2 (Vehicle data bus non-activity triggered Sleep). The command "AT PC" makes IC not to detect signal from vehicle data bus. After the sleep delay time,DFL168A will display

```
Sleep
```

and then delay 20 to 21 ms, PD pin will output sleep Logic level, then DFL168A sleeps. Users will have enough time to do something before users shutdown itself using PD pin.

The only way to wake up IC is to type key "U" in hyperterminal, DFL168A will wake up and PD pin will output Normal Logic level and delay 4 to 5 ms and display:

```
IC wakes up
>
```

IC will restore state which is one of before sleep.
After wake up, you must send "AT TP h" or "AT SP h" command. Otherwise DFL168A will sleep again.

The power consumption in Method 1 and Method 2 when IC enters sleep is the least. The power consumption in Method 3 is more than one of Method 1 and Method 2. However, Method

3 can use software to control.

The command "AT SLEEP PIN 1" will select method 1 (Sleep-pin triggered Sleep)
The command "AT SLEEP PIN 0" will select method 2 and 3 (Vehicle data bus non-activity triggered Sleep,Software Command triggered Sleep)
The command "AT SLEEP hhhh" will set sleep delay time to hhhh(Hex) seconds.
The command "SLEEP P 1" will make sleep pin to be active in logic 1.
The command "SLEEP P 0" will make sleep pin to be active in logic 0.
The command "AT PD 1" will make PD pin to output logic 1 when sleep.
The command "AT PD 0" will make PD pin to output logic 0 when sleep.
The command "AT PD 2" will disable PD pin output , PD pin becomes VF pin

6.7 IC Self-Diagnosis

DFL168A has self diagnosis function. It is useful for troubleshooting.
When IC enters self-diagnosis state, Sleep is disabled automatically.
The Command "AT SelfD 0" or ""AT SelfD 1" will make DFL168A enters self-diagnosis state.
The Command "AT SelfD 1" will make Dout0 to alternate between 0 and 1 every 500ms when self-diagnosis.
The Command "AT SelfD 0" will disable Dout0 output when self-diagnosis.

DFL168A will give a 32 bits' DTC when it competes diagnosis.
The below is an example:
(We use C++ style to comment each AT command and results even though DFL168A does not support comments)

```
>AT SELFD 0          //start diagnosis
IC's DTC is
00000000             // 32 bits' DTC
IC's Inputs are      // The following contents will be displayed every 1 second
06A0                //A/D conversion result of Ain0
1 0 0 0             //First Hex is Din0 input, 2nd Hex is Din1 input, 3rd Hex is Din2 input,
4th Hex is Din3
06A0                //A/D conversion result of Ain0
1 0 0 0             //First Hex is Din0 input, 2nd Hex is Din1 input, 3rd Hex is Din2 input,
4th Hex is Din3
06A0               //A/D conversion result of Ain0
1 0 0 0             //First Hex is Din0 input, 2nd Hex is Din1 input, 3rd Hex is Din2 input,
4th Hex is Din3
06A0                 //A/D conversion result of Ain0
```

```
 1 0 0 0              //First Hex is Din0 input, 2nd Hex is Din1 input, 3rd Hex is Din2 input,
4th Hex is Din3
 06A0                  //A/D conversion result of Ain0
 1 0 0 0              //First Hex is Din0 input, 2nd Hex is Din1 input, 3rd Hex is Din2 input,
4th Hex is Din3
 06A0                   //A/D conversion result of Ain0
 1 0 0 0              //First Hex is Din0 input, 2nd Hex is Din1 input, 3rd Hex is Din2 input,
4th Hex is Din3
............
```

After displaying DTC, DFL168A will display "A/D conversion result of Ain0" and "Discrete Input" every 1 second unless you press any key to exit diagnosis mode.

32 bits' DTC contains diagnosis result.
The definition of DTC is shown below:
D0 (LSb) : 1 means TXD1 send 0 fault.
D1: 1 means TXD1 or TXD2 or RTS2 always 0 fault.
D2: 1 means TXD2 send 0 fault
D3: 1 means RTS2 send 0 fault
D4: 1 means RXD2 always 0 fault
D5: 1 means CTS2 always 0
D6: 1 means RXD2 always 1 fault
D7: 1 means CTS2 always 1 fault
D8: 1 means J1708 always 1 fault
D9: Reserved
D10: 1 means J1708 always 0 fault
D11: 1 means CAN BUS configuration fault
D12: 1 means CAN BUS transmit fault
D13: 1 means CAN BUS receive fault
D14: 1 means No slave one-wire device found
D15: 1 means one-wire CRC Error fault
D16 to D31 (MSb): Reserved.

For DFL168A IC, D0 to D7 of DTC are not available. They are only for DFL168A-module. Just ignore the results of D7 to D0 of DTC if you use DFL168A chips.
If DTC is 00000000 or (DTC & 0xFFFFFF00) is 00000000 (if you only use IC), then DFL168A is good.
If D5 is 1 for DFL168A-module, that does not mean fault occurs, it perhaps a device, which has logic 0 output ,connects CTS2. You have to identify this situation by yourself.
Similarly, if D15 is 1, it does not mean the one-wire is malfunction, it perhaps you didn't connect I-button.

When DTC is 0, the VBUS_Active pin will alternate between logic1 and 0 every 500ms. And when DTC is not 0, the VBUS_Active pin will alternate between logic1 and 0 every 250ms. This purpose is for display diagnosis result in case TXD1 fault.

Notes: *1 Diagnosis has some side-effect on Dout0. If Dout0 is logic 0 before entering diagnosis state, Dout0 will has 3 logic 1 pulse output. The first Logic1 pulse width is less than 30us. After the first logic 1 pulse disappears 800us, the second logic 1 pulse occurs, which width is less than 15us. After the second logic 1 pulse disappears 800us, the third logic 1 pulse occurs, which width is less than 20us.*

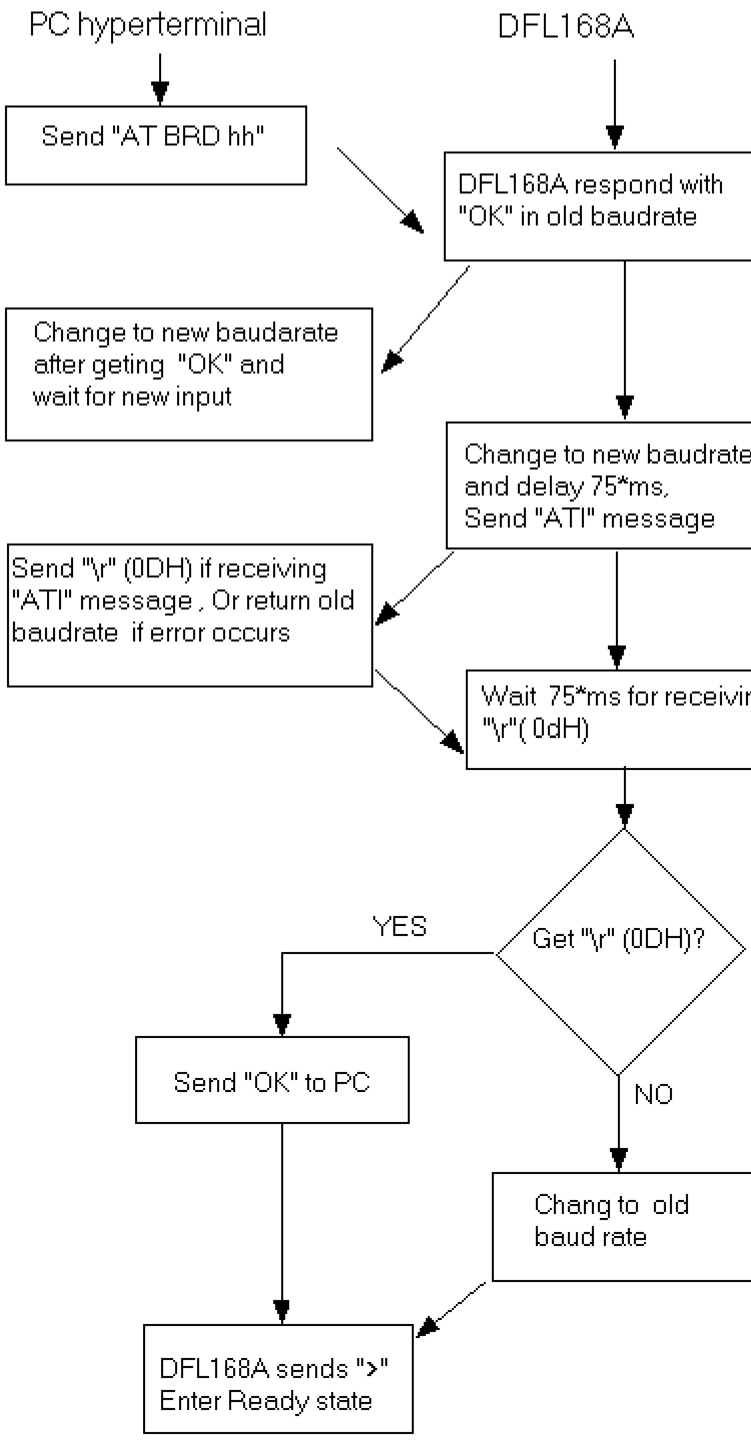
If Dout0 is logic 1 before entering diagnosis state, Dout0 will has 1 logic 0 pulse output. The Logic 0 pulse width is less than 18us.

If your device which connect Dout0 can't stand this pulse, please disconnect it before diagnosis or don't use the diagnosis command.

2 The command "AT SelfD 1" will cause Dout0 alternate between logic 1 and 0 every 500ms, You can connect Dout0 to every discrete input DinX to check discrete inputs. However , if your DOUT0 connects a actual device, please don't use "AT SelfD 1", it maybe damage your device except you don't care.

6.8 Higher Baud Rate for UART1

UART1 can use as high as 5Mbps of baud rate. However, success will depends on hardware.
In order to use new baud rate, the command "AT BRD hh" has a test process. If new baud rate passes successfully, UART1 will use new baud rate, otherwise, old baud rate will be used. The test process is shown below:


PC hyperterminal
DFL168A
Send "AT BRD hh"
DFL168A respond with "OK" in old baudrate
Change to new baudarate after geting "OK" and wait for new input
Change to new baudrate and delay 75*ms, Send "ATI" message
Send "\r" (0DH) if receiving "ATI" message , Or return old baudrate if error occurs
Wait 75*ms for receiving "\r"(0dH)
Get "\r" (0DH)?
YES
NO
Send "OK" to PC
Chang to old baud rate
DFL168A sends ">" Enter Ready state

* 75ms time can be changed by "AT BRT hh"

6.9 AT Command Details

We will describe every AT commands DFL168A supports in current version.

The symbol "hh" below denotes hexadecimal digit.

The symbol "xx" below denotes hexadecimal digit.

The symbol "yy" below denotes hexadecimal digit.

The symbol "zz" below denotes hexadecimal digit.

The symbol "dddd " below denotes decimal digit "dd.dd"

All AT commands are shown below:

AD 0 and **AD 1** [Disable ISO15765 Mixed address*, Enable ISO15765 Mixed address]

This command controls whether ISO15765 use mixed address mode. Default is AD 0 (Disable)

ADC 0 and **ADC 1** [Disable J1939 address Claim*, Enable J1939 address claim]

This command controls whether IC does J1939 address claim when IC starts with J1939 Protocol. Default is ADC 0 (Disable)

ADT hh [Mixed address is set to hh, default is 80]

This command will set mixed address to hh when the mixed address mode is enabled. It is only for ISO15765 protocol.

AR [Automatically set the receive address]

After you executes this command, DFL168A will set up ID of received data packet automatically after every request to vehicle. This is default value. Some commands which set Filter/mask will cancel the function, you should use it to resume. This command is only for ISO15765 protocol.

BRD hh [try Baud Rate Divisor hh]

This command is used for changing the baud rate of UART1 temporally. The actual baud rate is 5000/hh Kbps. For example, we want 115.2Kbps of UART1. The divisor will be 5000/115.2= 43.4 (decimals) =2B (hex), So we use AT command:

```
>AT BRD 2B
```

The maximum of baud rate for UART1 will be 5Mbps (If your hardware is allowable). Many interface circuit is not allowed so big baud rate that there is a test procedures in DFL168A before the baud rate is used actually. Please read 6.8 for details. If you set divisor to 00, the baud rate will be 9600 bps. This AT command only temporally change the baud rate. If reset DFL168A, it will use the default baud rate. The default baud rate is 56700bps, which can be changed by PP 1C.

Note: *if you change the default baud rate of UART1, and you forget the value, you will not access DFL168A by UART1. The only solution is restoring the default baud rate. Firstly connect Ain0 to ground before power on DFL168A, then power on DFL168A, now the default baud rate will become 57600bps, please change the default baud rate by PP 1C.*

BRT hh [set Baud Rate Time out to hh]

This command sets up the timeout used for "AT BRD hh" baud rate test procedures. The time out will be hh x 5.0 ms. the default value is 75ms. It will be default 75ms when hh =00

CAF 0 and **CAF 1** [CAN Aoto formating PCI Off or on*]

This command is only for ISO15765 and it will decide whether DFL168A sets up PCI byte or bytes automatically for you. Default is auto (CAF 1). However, if flow control is on (after execute "AT CFC 1"), DFL168A still sets up PCI byte or bytes automatically even though you execute "AT CAF 0". So If you want to use your PCI, you should execute "AT CAF 0" and "AT CFC 0" commands.

CF hhh [set the CAN ID Filter to hhh]

The CAN Filter works in conjunction with the CAN Mask to decide what data packet is to be accepted by the receiver. As each message is received, the incoming CAN ID bits are compared to the CAN Filter bits (when the mask bit is a "1"). If all of the relevant bits match, the message will be accepted, and processed by the DFL168A, otherwise it will be discarded. This command sets up 11 bits ID of CAN received. Only the right-most 11 bits of "hhh" are used. This command cannot be used by "J1939/J1708"

CF hh hh hh hh [set the CAN ID Filter to hhhhhhhh]

This command sets up 29 bits ID of CAN received. Only the right-most 29 bits of "hh hh hh hh" are used. This command cannot be used by "J1939/J1708"

CFC 0 and **CFC 1** [CAN Flow Control Off or On*]

This command is for ISO15765 Protocol. The ISO15765 protocol expects a "Flow Control" message to always be sent in response to a "First Frame" message. DFL168A automatically sends "Flow Control" message without any intervention by the user if flow control is on (AT CFC 1). If experimenting with a non-OBD system, it may be desirable to turn automatic response off, and the "AT CFC 0" command has been provided for that purpose. Default is "CFC 1" (Flow Control On).

CM hhh [set the CAN ID Mask to hhh]

There are tons of messages being transmitted in a CAN Bus system at any time. In order to limit what DFL168A accepts, there needs to be a system of filtering out the relevant ones from all the others. This is done by the filter, which works in conjunction with the mask. A mask is a combination of many bits that tell DFL168A which bits in the filter are relevant, and which bits can be ignored. The "1" in the mask bit tells DFL168A "must match" condition, while "0" tells DFL168A "don't care". This 3 digital "hhh" is used for 11 bits ID system. Only right-most 11 bits of "hhh" is used. This command cannot be used by "J1939/J1708"

CM hh hh hh hh [set the CAN ID Mask to hhhhhhhh]

This command is used to assign mask for 29 bits ID system This command cannot be used by "J1939/J1708"

CP hh [set the CAN Priority bits to hh]

This command is used to set the five left-most bits of 29 bit CAN ID for request data packet The other 24 bits is decided by "AT SH hh hh hh" command. This command cannot be used by "J1939/J1708"

CRA hhh [set the CAN Rx Addr to hhh]

Setting Filter and mask are complex. This command will set 11 bits ID for 11 bit CAN bus response. DFL168 will automatically set up filter and mask for you after this command. The right-most 11 bits of "hhh" are used. This command cannot be used by "J1939/J1708"

CRA hh hh hh hh [set the CAN Rx Addr to hhhhhhhh]

Setting Filter and mask are complex. This command will set 29 bits ID for 29 bit CAN bus response. DFL168 will automatically set up filter and mask for you after this command. The right-most 29 bits of "hh hh hh hh" are used. This command cannot be used by "J1939/J1708"

CV 0000 [restore the default Calibration Value]

This command will restore calibration value of analog input. The default value is 11units/volts, Ain0 ranges from 0.0Volts to 3.3 Volts.

CV dddd [Calibrate the voltage to dd.dd]

This command makes the current Analog input (Ain0) reading to be dd.dd (decimals). And keep the calibration to flash memory. So after you execute this AT command, if the Ain0 keeps constant, the response will be dd.d (decimals) after you execute "AT RV".

For example, if Ain0 is contant,

```
>AT RV

11.4
>AT CV 1200
OK

>AT RV
12.0V
```

```
>
```

D [set all to Defaults]

This command makes all options to factory defaults. It's the same values as ones after you power on the DFL168A.

And when you execute "ATD" command, DFL168A will execute actual writing flash memory operations.

D0 and **D1** [display DLC off(0)* or on(1)]

This command control whether display the number of data byte of CAN BUS (DLC) when header byte display (AT H1). Default is D0 (Off)

DEV1 AC1 [get AC1 Broadcast data of ACE controller (Device 1)]

All command with beginner DEV1 will be related to UART 2 (Device 1). This command will get the broadcast data of AC1 for ACE controller.

DEV1 AC2 [get AC2 Broadcast data of ACE controller (Device 1)]

All command with beginner DEV1 will be related to UART 2 (Device 1). This command will get the broadcast data of AC2 for ACE controller.

DEV1 AC3 [get AC3 Broadcast data of ACE controller (Device 1)]

All command with beginner DEV1 will be related to UART 2 (Device 1). This command will get the broadcast data of AC3 for ACE controller.

DEV1 ACE DIST [get the Distance traveled of ACE controller (Device 1)]

We know the Distance traveled of ACE controller is the distance traveled between 2 events of AC3 broadcast. If you parse the data of AC3 broadcast to get total distance traveled, but you missing some AC3 broadcasts, you will got wrong total distance traveled (less than the actual value). DFL168A automatically accumulates the distance traveled when event of AC3 broadcast occurs. So you use this command to get total distance traveled. Never worry about missing data.

DEV1 ACE DIST hhhh [set the initial Distance traveled to hhhh]

DFL168A automatically accumulates the distance traveled when event of AC3 broadcast occurs. This command is used for setting the initial value of distance traveled to hhhh (Hex) , For example,

```
>AT DEV1 ACE DIST 0000
OK

>AT DEV1 ACE DIST
0000
```

```
>
```

DEV1 ACE QtyOfS [get the spread Quantity of Solid material from ACE controller (Device 1))]
It is almost the same as "AT DEV1 ACE DIST" except the parameter is spread Quantity of Solid material.

DEV1 ACE QtyOfS hhhh [get the spread Quantity of Solid material from ACE controller (Device 1))]
It is almost the same as "AT DEV1 ACE DIST hhhh" except the parameter is spread Quantity of Solid material.

DEV1 ACE QtyOfL [get the spread Quantity of Liquid from ACE controller (Device 1))]
It is almost the same as "AT DEV1 ACE DIST" except the parameter is spread Quantity of Liquid.

DEV1 ACE QtyOfL hhhh [get the spread Quantity of Liquid from ACE controller (Device 1))]
It is almost the same as "AT DEV1 ACE DIST hhhh" except the parameter is spread Quantity of Liquid.

DEV1 ADR REQ hhhh
[send data Request to CS230 controller of device 1 in order to get data of Address hhhh]
This command is only used for CS230 controller. It will request parameter which map address is hhhh (Hex).

DEV1 ALT [get Altitude from GPS (Device 1)]
This command is used for getting the altitude from GPS (Device 1). We know that altitude is available for GGA sentence. So we have to get GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 2
OK

>AT DEV1 ALT
1080

>
```

It means the antenna of GPS is 1080 meters above mean-sea-level.

DEV1 BRD hh [set Baud Rate Divisor hh for UART2 (Device 1)]
All command with beginner DEV1 will be related to UART 2 (Device 1). This command is similar to "AT BRD hh". But the differences are the following items:

1. "DEV1 BRD hh" is for UART2
2. "DEV1 BRD hh" has no test procedures. it directly set up baud rate.
3. "DEV1 BRD 01" means baud rate is 4800Bps
4. "DEV1 BRD 02" means baud rate is 2400Bps
5. Maximum of UART2 baud rate will be 5000/3=1666.67KBps

This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 BT hh [set data packet Broken Time to hh (Device 1)]

In general, the time between bytes in a data packet is very short. If the time is too long, we think that the data packet is broken. This command is used for broken time of data packet for UART2. The broken time is hh x 4ms, default value is 0A (hex), which is 40ms. However, if hh is 00, the broken time is 200ms. You should make sure this value hh must be greater than hh in "AT DEV1 S hh" when both SOF and EOF are disabled. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 CLR [Clear the history data of UART 2 (Device 1)]

When UART2 receives broadcast data, DFL168A will put these data into history data buffers if DFL168A has no command of receiving data request. This command will clear all history data buffers.

DEV1 CS h [set Check Sum adding extra h]

when receiving data packet of UART2 uses check sum, we must use "AT DEV1 CSM hh" command to set up check sum mode. We will increase the check sum value by this "AT DEV1 CS h" command for every check sum mode. The additional value will be h (Hex). For example

```
>AT DEV1 CSM  01
OK
```

```
>AT DEV1 CS A
>OK
```

It means that receiving data packet will be valid if summing up all data bytes including Checksum itself is A (hex). Of cause, we have to discard the most significant bytes when calculating checksum. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 CS hh [set Check Sum adding extra hh]

It is the same as "AT DEV1 CS h" except the extra bytes is 2 hex data. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 CS hhhh [set Check Sum adding extra hhhh]

It is the same as "AT DEV1 CS h" except the extra bytes is 4 hex data. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 CSM hh [set Check Sum Mode hh (Device 1)]

Receiving data packet of UART2 may has different check sum method. This command is used for setting check sum mode of UART2 to hh (Hex) . We indicate checksum mode byte as B7 B6 B5 B4 B3 B2 B1 B0 (B7 is MSb, B0 is LSb).

B7------ 1: Check sum uses double bytes' word, 0:Check sum uses single byte
B6------- 1: First MSB for Checksum word, 0:First LSB for Checksum word
B5------- Reserved
B4------- Reserved
B3--------Reserved
B2 B1 B0----------- 0: No check sum
1: All bytes arithmetically sum up and at last inverse the number. (-SUM)
2: All bytes arithmetically sum up (SUM)
3: All bytes excluding the bytes before data packet length arithmetically sum up and at last inverse the number. (-SUM)
4: All bytes excluding the bytes before data packet length arithmetically sum up (SUM)
5:All bytes excluding the bytes before data packet length and excluding data packet length byte arithmetically sum up and at last inverse the number. (-SUM)
6:All bytes excluding the bytes before data packet length and excluding data packet length byte arithmetically sum up. (SUM)
7: Reserved

For example, if check sum is that all bytes including check sum itself addition will be zero, the check sum mode will be 2, and "AT DEV1 CS 00".

This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 DJ RD xx yy

[Dickey-John controller (Device 1) Read data from data group xx and data element yy]

This command is used for read data from Dickey-John controller (Device 1). What kind of data will be read? it is decided by Data group number xx (Hex) and Data Element number yy (Hex). If the data of read is binary, it will be displayed followed by "H".

DEV1 DJ RD xx yy zz

[Dickey-John controller (Device 1) Read data from data group xx and data element yy and data sub element zz]

This command is used for read data from Dickey-John controller (Device 1). What kind of data will be read? it is decided by Data group number xx (Hex) and Data Element number yy (Hex) and Data Sub

Element number zz (Hex). If the data of read is binary, it will be displayed followed by "H".

DEV1 DATE [get Date from GPS (Device 1)]

This command is used for getting the date from GPS (Device 1). We know that date is available for RMC sentence. So we have to get RMC broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 DATE
25/01/2012

>
```

It means 25th day of January 2012

DEV1 DP [Describe device1 Protocol by text]

This command is used for display current device of UART2 by text.

DEV1 DPN [Describe device1 Protocol by number]

This command is used for display current device of UART2 by number.

0 denotes "General Binary Device"

1 denotes "General ASCII Device"

2 denotes "CS440 controller"

3 denotes "CS230 controller"

4 denotes "Dickey-John Controller"

5 denotes "ACE Spreader Controller"

6 denotes "GPS with NMEA 0183 output"

DEV1 EOF [cancel the "End of Frame" byte for Device 1]

This command is used for telling DFL168A No tail byte of receiving data packet for UART2. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 EOF hh [set the "End of Frame" byte to hh for Device 1]

This command is used for setting up the tail byte of receiving data packet for UART2. The tail byte will be hh (Hex).This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 Exit T hh [set the "End of transparent mode" byte to hh for Device 1]

This command is used for setting up the a special character for exiting "Transparent mode of device1" . The default character is "ESC" (hh=0x1B).

"Transparent mode of device1" means that all transmit and receive of UART2 will be changed to "UART1". It looks like external device connects UART1 directly.
You can exit "Transparent mode of device1" by transmitting a special character to UART1 from outside. You can exit "Transparent mode of device1" by time span device 1 used during transparent mode, this time span is the value of command "AT DEV1 ST hh"

DEV1 GB [Get Broadcast data of Device 1, don't care which SOF it is]
This command is used for getting any broadcast data packet of Device 1 (UART2)

DEV1 GB SOF h [Get Broadcast data of Device 1 with the SOF number h]
This command is used for getting a special broadcast data packet of Device 1. The header (SOF) of data packet is SOF number h.
We can set up maximum of 5 different header (SOF) by "AT DEV1 SOF h xx ... mm" command

DEV1 HFC1 and **DEV1 HFC0**
[enable hardware flow control RTS2/CTS2 (1), or disable hardware flow control RTS2/CTS2 (0)]
This command is used for enabling flow control RTS2/CTS2 for UART2. The default depends on Device 1. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 HT1 and **DEV1 HT0**
[enable history data of UART 2 (Device 1), or disable history data of UART 2 (Device 1)]
Some device only broadcasts some data packets (for example , Device ID)once when power on. If you didn't enable history data and you didn't send "AT DEV1 GB" or "AT DEV1 GB SOF h" command on time, you will miss the data packet. However, DFL168A provides history data buffers, so receiving data packet will be put into history data buffers when user didn't send "AT DEV1 GB" or "AT DEV1 GB SOF h" command on time. If we enable history data of UART 2 ("AT DEV1 HT1"), IC will get data packet from UART2 firstly. And then if fails to get data packet from UART2, it will get data packet from history data buffers.

DEV1 Lat [get Latitude from GPS (Device 1)]
This command is used for getting the latitude from GPS (Device 1). We know that latitude is available for RMC or GGA sentence. So we have to get RMC or GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 LAT
+25 33.456'
```

```
>
```

It means latitude is north 25 degree 33.456 minutes

DEV1 LEN [cancel the receiving data packet length of device 1]

This command is used for telling DFL168A No length of receiving data packet for UART2. How to separate data packet if no length? It will use the time separation between bytes or use the tail byte. This command must execute under Dev1 protocol is close by "AT DEV1 PC".

DEV1 LEN hh [set the receiving data packet length of device 1 to fixed value hh]

This command is used for setting up the receiving data packet length of UART2 to fixed value hh (Hex). The length of data packet means all bytes including header bytes (SOF) and tail byte (EOF). This command must execute under Dev1 protocol is close by "AT DEV1 PC".

DEV1 LEN xx hh

[set the receiving data packet length of device 1 to a value, which is the byte located in xx]

Some receiving data packets have variable data packet length. But the length information is known, and it is content of a receive byte. This command dictates the length information is in the xx (Hex) position of receiving byte. The number of receiving byte position starts from number 0 instead of number 1. Our length definition is all bytes including header bytes (SOF) and tail byte (EOF). But for some devices, the length is not including header bytes (SOF) and/or tail byte (EOF). In order to adjust the length to meet our definition, we will add hh (Hex) to make sure that the length includes header bytes (SOF) and tail byte (EOF). This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 Long [get Longitude from GPS (Device 1)]

This command is used for getting the longitude from GPS (Device 1). We know that longitude is available for RMC or GGA sentence. So we have to get RMC or GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 2
OK

>AT DEV1 LONG
-105 33.4567'

>
```

It means longitude is west 105 degree 33.4567 minutes

DEV1 PC [Device 1 Protocol Close]

This command is used for stopping Device 1. After executing this command, UART2 will not receive any message and transmit any message. "AT DEV1 SP h" or "AT DEV1 TP h" or "AT DEV1 REQ" or "AT DEV1 REQ hh hh ... hh" or "AT GB" or "AT GB SOF h" or "DEV1 ADR REQ hhhh" or "AT DEV1 DJ RD xx yy" or ""AT DEV1 DJ RD xx yy zz" or Silence Request command will start UART2 again.

DEV1 REQ [send data Request to CS440 controller of device 1]

This command is used only for CS440 controller. It will send a request to CS440 controller. CS440 will respond with 86 bytes of data.

DEV1 REQ hh hh ... hh

[transmit Request hh hh ...hh to Device 1. The meaning is different for ASCII Device and Binary Device]

This command is used for sending request to Device 1 (UART2). For binary device, hh hh ... hh (Hex) will be hexadecimal sequence. The quantity of hex must be even. Device 1 will receive the binary request sequence hh hh ... hh (Hex). And Device1 will respond to it in binary sequence (Of cause, display hex to us).

For ascii device, hh hh ... hh will be ascii characters, The quantity of characters may be any except 0. And characters are case sensitive. Some characters such as space must use escape character. DFL168A supports the following escape characters:

Escape character	Hexadecimal value
\0	00
\n	0A
\r	0D
\t	09
\v	0B
\a	07
\f	0C
\'	27
\"	22
\?	3F
\\	5C
\xhh	hh

Note: "hh" in "\xhh " will be hexadecimal number. In this way, we can transmit any ascii character. Some characters in above escape character table can be directly typed it, for example, \', \", \? can use ' and " and ? directly.

For ascii device response, the ascii characters will be displayed directly.

DEV1 RSOF h [set Response SOF of Device1 to number h]

This command is used for setting which response will be received in UART2. It is decided by which header (SOF) of data packet. So h (Hex) will be header (SOF) number of data packet.
In general , we must set up response header (SOF) before sending request command of "AT REQ hh hh ... hh".

DEV1 S [cancel the separating time between the receiving data packets]
This command is used for telling DFL168A no separating time between receiving data packet for UART2. How to separate data packet if no separating space? It will use the length of data packet or use the tail byte. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 S hh [set the separating time hh between the receiving data packets]
This command is used for setting up the separating time between receiving data packet for UART2 to hh x 4 ms. Default value is 14 (Hex), that is 80ms. However, if hh is 00, the separating time is 200ms. You should make sure this value hh must be less than hh in "AT DEV1 BT hh" when both SOF and EOF are disabled. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 SI GB [get silence broadcast of Device 1]
This command is the same as "AT DEV1 GB" except hiding the response results. This silence command is used for letting DFL168A get data, and then you can use parse command to get parameters, such as "AT DEV1 LAT" ,..., and "AT DEV1 TIME"

DEV1 SI GB SOF h
[Get Broadcast data of Device 1 with the SOF number h. The response does not display]
This command is the same as "AT DEV1 GB SOF h" except hiding the response results. This silence command is used for letting DFL168A get data, and then you can use parse command to get parameters, such as "AT DEV1 LAT" ,..., and "AT DEV1 TIME"

DEV1 SI REQ [send Silence data Request to CS440 controller of device 1]
This command is the same as "AT DEV1 REQ " except hiding the response results. This silence command is used for letting DFL168A get data, and then you can use parse command to get parameters.

DEV1 SI REQ hh hh ... hh [get Silence Request of Device 1]
This command is the same as "AT DEV1 SI REQ hh hh ... hh" except hiding the response results. This silence command is used for letting DFL168A get data, and then you can use parse command to get parameters, such as "AT DEV1 LAT" ,..., and "AT DEV1 TIME"

DEV1 SOF h [cancel the SOF of Number h]

This command is used for for telling DFL168A no header (SOF) of Number h for receiving data packet for UART2. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 SOF h xx ... xx [set the SOF of Number h to xx ... xx]

This command is used for setting up the header (SOF) of Number h for receiving data packet for UART2. The SOF of Number h will be xx ... xx (Hex). The maximum length of SOF is 6 bytes. The SOF numbers are 1 to 5. This command must execute under Dev1 protocol is close by "AT DEV1 PC"

DEV1 SP h [set Device 1 Protocol h and save it]

This command is used for setting the device type of UART2. And it will store in flash memory. Next time when power on, it will automatically use the device for UART2. Currently, DFL168A supports the following devices:

0 : "General Binary Device"
1 : "General ASCII Device"
2 : "CS440 controller"
3 : "CS230 controller"
4 : "Dickey-John Controller"
5 : "ACE Spreader Controller"
6 : "GPS with NMEA 0183 output"

And when you execute "AT DEV1 SP h" command, DFL168A will execute actual writing flash memory operations.

DEV1 SPEED [get Speed from GPS (Device 1)]

This command is used for getting the speed of vehicle from GPS (Device 1). We know that speed is available for RMC or VTG sentence. So we have to get RMC or VTG broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 3
OK

>AT DEV1 SPEED
56.7

>
```

It means the speed is 56.7 knots

DEV1 ST hh [Set Device 1 response Timeout to hh]

When you send a request to device 1 by UART2, DFL168A will wait for response from device 1. If DFL168A gets response, it will display the result to user via UART1. If DFL168A does not get response from device1 with the specified time, it will display "No Response" via UART1. The specified time is set by this command.

The specified time is hhx4 ms when hh is less than FA (Hex) except 00.

The specified time is 500 ms when hh is 00.

The specified time is 2 seconds when hh is FB (Hex).

The specified time is 4 seconds when hh is FC (Hex).

The specified time is 6 seconds when hh is FD (Hex).

The specified time is 8 seconds when hh is FE (Hex).

The specified time is 10 seconds when hh is FF (Hex).

Default hh is different for different Device 1. Please read 6.5 Device Access

DEV1 TIME [get Time from GPS (Device 1)]

This command is used for getting the time from GPS (Device 1). We know that time is available for RMC or GGA sentence. So we have to get RMC or GGA broadcast of GPS before sending this command. For example,

```
>AT DEV1 SI GB SOF 1
OK

>AT DEV1 TIME
20:58:31

>
```

Time display format is hh:mm:ss

DEV1 TP h [Try device 1 Protocol h and no save it]

This command is the same as "AT DEV1 SP h" except it does not save the setting. We recommend to use it instead of "AT DEV1 SP h" because it can decrease times of erase/writing flash.

DEV1 TPK 1/0 [Enable (1) /Disable (0) special character of ending "Transparent mode"]

This command is used for enabling/disablling the a special character for exiting "Transparent mode of device1" . The default is enabled.

"Transparent mode of device1" means that all transmit and receive of UART2 will be changed to "UART1". It looks like external device connects UART1 directly.

DEV1 TPT 1/0 [Enable (1) /Disable (0) timeout of ending "Transparent mode"]

This command is used for enabling/disablling the time span for exiting "Transparent mode of device1" . The default is enabled.

"Transparent mode of device1" means that all transmit and receive of UART2 will be changed to

"UART1". It looks like external device connects UART1 directly.

DEV1 TransP [Enter "Transparent mode" for device 1]

This command will let device 1 enter "Transparent mode of device1" .

"Transparent mode of device1" means that all transmit and receive of UART2 will be changed to "UART1". It looks like external device connects UART1 directly.

You can exit "Transparent mode of device1" by transmitting a special character to UART1 from outside. You can exit "Transparent mode of device1" by time span device 1 used during transparent mode, this time span is the value of command "AT DEV1 ST hh"

DM1 [Monitor for DM1 messages]

This command is only for J1939 protocol. It continuously monitors the broadcast of diagnostic mode 1 message, and displays results via UART1. It will exit monitor mode when UART1 receive any character.

DP [Describe the current Protocol by text]

It will return the current protocol of vehicle bus via UART1. The protocol displayed is text.

Currently, DFL168A supports ISO15765, User1, User2, J1939 and J1708/J1587 protocols.

DPN [Describe the current Protocol by Number]

It will return the current protocol of vehicle bus via UART1. The protocol displayed is hexadecimal number.

Currently, DFL168A supports ISO15765, User1, User2, J1939 and J1708/J1587 protocols.

It will display "6" for 11bits ISO15765 with 500kbps baud rate, and "7" for 29 bits ISO15765 with 500kbps baud rate, and "8" for 11 bits ISO15765 with 250kbps baud rate, and "9" for 29 bits ISO15765 with 250kbps baud rate, and "A" for J1939 protocol, and "B" for 11 bits User1 CAN with 125kbps baud rate, and "C" for 11 bits User2 CAN with 50kbps baud rate, and "D" for J1708/J1587 protocol

E0 and **E1** [Echo Off, or On*]

It is for UART1. UART1 will re-transmit the character it receive when E1 (Echo on). This is convenient to users who use the hyperterminal, they will can see what they type. However, for interface with UART1 via software, they do not need to see what the software transmits, so turn off echo by E0 command.

FCSD [1-5 bytes] [Flow control Set Data to ...]

The data bytes of flow control are set by this command when flow control mode is 1 or 2. You can use this command set 1 to 5 bytes of flow control data. It is only for ISO15765 Protocol.

FC SH hhh [Flow control Set Header to ...]

The 11 bits ID of flow control is set for 11 bits ISO 15765 by this command when flow control mode is 1. Only right-most 11 bits of "hhh" are used. It is only for ISO15765 Protocol.

FC SH hh hh hh hh [Flow control Set Header to ...]

The 29 bits ID of flow control is set for 29 bits ISO 15765 by this command when flow control mode is 1. Only right-most 29 bits of "hh hh hh hh" are used. It is only for ISO15765 Protocol.

FC SM h [Flow control Set mode to h]

The command sets how to respond to First Frame message when automatic flow control is enabled. The "h" can be "0" (mode 0) or "1"(mode 1) or "2"(mode 2). Mode "0" means that DFL168A automatically respond to First Frame message, user will do nothing. Mode "1" means that DFL168A automatically respond to First Frame message, but user must prepare 11 bits or 29 bits ID by "AT FC SH" command and prepare flow control data by "AT FC SD ..." command. Mode "2" means that DFL168A automatically respond to First Frame message, but user must prepare flow control data by "AT FC SD ..." command. Of cause, Automatic respond to First frame must be under the condition "Flow control is enabled" (AT CFC 1). Default is mode 0

H0 and **H1** [response header off*, or header On]

These commands turn off/on header bytes display in UART1. Those header bytes can come from response of J1939 ISO15765 USER and J1708 and device 1 (SOF). In general, we turn off header bytes display by H0. Default is H0. If we want to see more details of response, we can see header by H1. However we can only see header and length byte, can't see the CRC and Check Sum and tail byte (EOF). There is an exception, we can see SOF and Check Sum and EOF for GPS device when H1

I [print the version ID]

Identify itself. It will display IC information. For example,

```
 >ATI
DFL168A V1.10

>
```

INTRUDE0 and **INTRUDE1**

[transmit message into truck data bus Off, transmit message into truck data bus On*]

These commands control whether or not the user's request data are actually transmitted to vehicle data bus (J1939, J1708). In general, truck will broadcast messages via data bus. So users do not need to transmit request to truck data bus because any message transmitted will occupy the bandwidth of truck data bus. In this situation, we can send command "AT INTRUDE 0" to disable transmit. Of cause, we can get the message we want even though we didn't send request because truck broadcast periodically. On the other side, some messages are not broadcast. it will send when getting request. So for these messages, we can temporally turn on transmission by "AT INTRUDE 1", and then send request command. After we get response, we can turn off transmission again. Default is "Intrude 1".

Note: *ISO15765 and User1/user2 protocols are not broadcast protocols, so it will send request even though you execute "AT INTRUDE 0"*

JB and JE [J1939 request PGN data use big endian*]

These commands are for J1939 PGN request. A PGN is 3 byte's data (total 18 bits). It can used 2 bytes' data if we ignore page number. Our habit is MSB first LSB last when we write a multi-bytes data. However, J1939 protocol is MSB last LSB first for multi-bytes data. If we use JB or JE command, we will keep our habit when sending a request. Default is JB or JE. For example, we request PGN :61444 (decimals) F004(Hex), type the following request

```
 >F004
FF FF A2 00 7D FF FF FF

>
```

Note: Response from J1939 still be MSB last LSB first even though you use JB or JE command

JL and JS [J1939 request PGN data use little endian]

These commands are for J1939 PGN request. It will keep the habit of J1939 when request PGN. J1939 protocol is MSB last LSB first for multi-bytes data. For example, we request PGN :61444 (decimals) F004(Hex), type the following request

```
>04F0
FF FF A2 00 7D FF FF FF

>
```

L0 and **L1** [Linefeeds Off, or On*]

These commands control whether or not response messages to UART1 add a line feed character after each carriage return character. For L0, it is not added line feed character. For L1, it is added line feed character. Default is L1. you can change the default by PP 00

MA [Monitor All]

This command is for monitoring all messages on the vehicle data bus. For J1708/J1587, "AT H0" and "AT H1" will have impact on this command. However, for J1939, all PGNs display will contain header because it makes no sense if missing header for J1939 when monitoring all PGNs. DFL168A will exit monitor mode when UART1 receive any character.

MMID 00 [Monitor all MID]

This command is only for J1708/J1587. It will monitor any MID on the truck bus. It is almost the same as "AT MA".

The difference is that "MMID 00" will combine the multisection parameter to one over-sized data packet and "MA" will display what it gets. DFL168A will exit monitor mode when UART1 receive any character.

MMID xx and **MMID xx 00** [Monitor MID xx]

This command is only for J1708/J1587. It will monitor specified MID on the truck bus. The specified MID will be xx (Hex).

DFL168A will exit monitor mode when UART1 receive any character.

MMID xx hh [Monitor MID xx with PID hh]

This command is only for J1708/J1587. It will monitor specified MID and specified PID on the truck bus. The specified MID will be xx (Hex). The specified PID will be hh (Hex) and hh is not 00. DFL168A will exit monitor mode when UART1 receive any character.

MMID xx hh ... hh [Monitor MID xx with mutiple different PID hh ... hh]

This command is only for J1708/J1587. It will monitor specified MID and more specified PIDs on the truck bus. The specified MID will be xx (Hex). The specified PIDs will be at most 8 bytes' hh (Hex). All PIDs can not be 00. The maximum of PIDs monitored is 8. DFL168A will exit monitor mode when UART1 receive any character.

MP hhhh [Monitor for PGN hhhh (ignore priority and page number)]

This command is only for J1939. It will monitor specified PGN and ignore the page number and priority. The specified PGN will be hhhh (Hex). Byte order is decided by "AT JB" or "AT JE", or "AT JL" or "AT JS". DFL168A will exit monitor mode when UART1 receive any character.

MP hhhhh [Monitor for PGN hhhhh (only ignore priority)]

This command is only for J1939. It will monitor specified PGN and only ignore priority. The specified PGN will be hhhhh (Hex). Byte order is decided by "AT JB" or "AT JE", or "AT JL" or "AT JS". DFL168A will exit monitor mode when UART1 receive any character.

MR hh [Monitor for Receiver hh]

This command is similar to the "AT MA" command except that it will display messages that are sent to the hex address hh. This command is only for ISO15765

MT hh [Monitor for Receiver hh]

This command is similar to the "AT MA" command except that it will display messages that are sent by transmitter hex address hh. This command is only for ISO15765

OW RD [One-Wire Read]

This command will read one-wire slave I-button. It will have 7 bytes' data, First 6 bytes is Serial Number of I-Button, the last byte is family code , it should be 01 (Hex). If no slave device connected one-wire, you will see "No Device Connected" in hyperterminal. If slave device connected one-wire but CRC is wrong, you will see " CRC ERROR" in hyperterminal.

PC [Protocol Close]

This command will stop communication with vehicle data bus. It may cause the sleep of IC if sleep mode is bus-active related. When you execute "AT PC" command , the speed pin output will keep the same the pulse frequency as one before stop. The communication with vehicle data bus can be restored by request command.

PD1 and PD0 and PD2

[Power Down pin out logic1, or Power Down pin out logic 0, or Disable Power Down pin out]

These commands control power down pin output (PD). PD1 means the PD pin will output logic 1 when sleep. PD0 means the PD pin will output logic 0 when sleep. PD2 means the PD pin will be disabled and VF pin will be enabled.

User can use the PD pin to shut down other devices or ICs when sleep. Default is PD1, it can be changed by PP 1A PP 04

PP xx OFF [disable Programmable Parameter xx]

This command will disable the programmable parameter number xx (Hex). The parameter in the number xx will not be used.

The factory default will be used again. In general, we have to restart IC when we change PP and enable the parameter. We can execute "ATZ"or "AT@R" or re-power on to restart IC.

PP FF OFF [disable all Programmable Parameter]

This command will disable all programmable parameters. This is a factory default value.

PP xx ON [enable Programmable Parameter xx]

This command will enable the programmable parameter number xx. The parameter in the number xx will be used.

In general, we have to restart IC when we change PP and enable the parameter.

PP FF ON [enable all Programmable Parameter]

This command will enable all programmable parameters. In general, we have to restart IC when we change PP and enable the parameter.

PP xx SV yy [for Programmable Parameter xx, Set the value to yy]

This command will change the value of programmable parameter number xx(Hex) to new value yy (Hex). If you want use it, you have to use "AT PP xx ON" to enable this parameter and restart IC.

PPS [print a Programmable Parameter Summary]

This command will display Programmable Parameters Summary in hyperterminal. Display format is xx: hh F/N where xx is parameter number and hh is hexadecimal value, F denotes OFF and N denotes ON.

PPP [all Programmable Parameters store into flash]

This command will write all programmable parameters into flash memory. We provide the command because we use flash to store parameters instead of EEPROM. Flash writing operation is page-based writing. So we hope to write parameters together.

R0 and **R1** [Response Off, or On*]

This command will control whether or not IC receives the response from vehicle data bus after sending request. R0 will refuse response. R1 will receive response.

RD h [Read discrete input h (Din h)]

This command will read input from Din Number h, the result will be 0 or 1. if you execute "AT DEBOUNCE1" before "AT RD h", the result of reading has deleted side-effect of switch debounce.

RTR [Send RTR message]

This command causes a special "remote frame" to be sent without any data. This command is only for non-format User protocol.

RV [Read analog input (value is decimals)]

This command will read analog input from Ain0. The result will be decimals. The display format is x.xV or xx.xV or xxx.xV. The data ranges from 0.0V to 999.9V

S0 and **S1** [printing of space Off, or On*]

These command will control whether or not space character is inserted between bytes of response. S1 is default. It will insert space character between bytes for human good reading. S0 will cancel the space character, it is ok for machine reading.

SelfD 0 [start IC self-Diagnosis without discrete out0 output]
This command will start IC self-diagnosis. And it won't make out0 alternates between 0 and 1. Diagnosis process can exit by any character from hyperterminal.

SelfD 1 [start IC self-Diagnosis with discrete out0 alternating 0 and 1 output]
This command will start IC self-diagnosis. And it will make out0 alternates between 0 and 1 in 500ms period after completing diagnosis. Diagnosis process can exit by any character from hyperterminal.

SH hhh [Set the header to hhh]
This command will set the 11 bit ID of request for 11 bits CAN Bus system. Only the right-most 11 bits of "hhh" are used. It cannot be used in J1939/J1708.

SH hh hh hh [Set the right-most 24 bits of 29 bits header to hh hh hh]
This command will set the right-most 24 bits ID of request for 29 bits CAN Bus system. The Left-most 5 bits of 29 bits ID are decided by "AT CP hh" command. It cannot be used in J1939/J1708

SLEEP hhhh [set Sleep delay time to hhhh seconds]
This command will set the sleep delay to hhhh(Hex) seconds. If sleep signal is active, and it is still active after hhhh(Hex) seconds, DFL168A will sleep.

SLEEP P1 and **SLEEP P0** [Sleep pin Polarity is logic-1 active, or Sleep pin Polarity is logic-0 active*]
These command will control sleep pin polarity. "SLEEP P1" means logic 1 will be active for sleep signal. "SLEEP P0" means logic 0 will be active for sleep signal. Default is logic 0 active.

SLEEP PIN1 and **SLEEP PIN0** [Sleep Pin enable, or Sleep Pin disable*]
These command will control sleep pin enabled/disabled. "SLEEP PIN1" means sleep pin will be enabled. And trigger level is decided by "SLEEP P1 or 0" command. "SLEEP PIN0" means sleep pin will be disabled. This pin becomes discrete input DIN3, and sleep signal will be from vehicle data bus. If IC didn't receive any message from vehicle data bus in longer than 5 seconds, the sleep signal will be triggered. And actual sleep will occur in some seconds which is decided by "SLEEP hhhh" command. Default is Sleep Pin disable.

SP h [Set current Protocol to h and save it]
This command will set the vehicle protocol to h (Hex) and save it. So IC will use it even though power off. Currently, DFL168A supports J1708/J1587 ISO 15765, J1939 and USER1/USER2.

If h is 0, the vehicle protocol will be auto. It means that it will search a proper ISO15765. J1939/J1708/ USER does not support auto.
If h is 6, the vehicle protocol will be ISO15765 with 11 bits ID and 500Kbps baud rate.
If h is 7, the vehicle protocol will be ISO15765 with 29 bits ID and 500Kbps baud rate.
If h is 8, the vehicle protocol will be ISO15765 with 11 bits ID and 250Kbps baud rate.
If h is 9, the vehicle protocol will be ISO15765 with 29 bits ID and 250Kbps baud rate.
If h is A, the vehicle protocol will be J1939.
If h is B, the vehicle protocol will be USER1 (Default: 11 bits ID CAN with 125Kbps baud rate, transmitted data length is not fixed, receive data ID: 11 bits and 29 bits).
If h is C, the vehicle protocol will be USER2 (Default: 11 bits ID CAN with 50Kbps baud rate, transmitted data length is fixed 8, receive data ID: only 11 bits).
If h is D, the vehicle protocol will be J1708/J1587. The other value of h will be reserved
Factory default is A. And when you execute "AT SP h" command, DFL168A will execute actual writing flash memory operations.

SP Ah [Set current Protocol to auto h and save it]
This command is similar to "SP h" except that it will search ISO15765 protocols when protocol h fails.

ST hh [Set response Timeout to hh]
When you send a request to vehicle by vehicle data bus, DFL168A will wait for response from vehicle. If DFL168A gets response, it will display the result to user via UART1. If DFL168A does not get response from vehicle with the specified time, it will display "No Data" via UART1. The specified time is set by this command.

The specified time is hh x 4 ms when hh is less than or equal to FA (Hex) except 00.
The specified time is 500 ms when hh is 00.
The specified time is 2 seconds when hh is FB (Hex).
The specified time is 4 seconds when hh is FC (Hex).
The specified time is 6 seconds when hh is FD (Hex).
The specified time is 8 seconds when hh is FE (Hex).
The specified time is 10 seconds when hh is FF (Hex).

Default hh is 32 (Hex) that is 200ms . Please read 6.4 Vehicle Bus Access

TP h [Try current Protocol to h (not save)]
This command is almost the same as "SP h" except no saving.

TP Ah [Try current Protocol to auto h (not save)]
This command is almost the same as "SP Ah" except no saving.

VF1 and **VF0** and **VF2**

[Vehicle Forward pin logic 1 out*, or Vehicle Forward pin logic 0 out, or Disable Vehicle Forward pin out.]

These commands control vehicle forward/backward pin output (VF). VF1 means the VF pin will output logic 1 when vehicle go forward. VF0 means the VF pin will output logic 0 when vehicle go forward. VF2 means the VF pin will be disabled and PD pin will be enabled. Default is VF2. It can be changed by PP 04 and PP 1A

WD 0 1 and **WD 0 0** [Write Dout0 1, or Write Dout0 0]

These commands control discrete output Dout0. "WD 0 1" will make Dout0 output logic 1. "WD 0 0" will make Dout0 output logic 0.

Z [reset all]

All parameters change to value in programmable parameter table, and execute reset operation, just like restart. When you execute "ATZ" command, DFL168A will execute actual writing flash memory operations before reset IC.

@R [reset all]

This command is the same as "ATZ" command.

@1 [display the IC description]

When you send "AT@1", you will see the "J1708/J1939/Spreader to RS232 Interpreter" information in your hyperterminal.

```
>AT@1
J1708/J1939/Spreader to RS232 Interpreter

>
```

@2 [display copyright]

This command will display Copyright information.

```
>AT@2
Dafulai Electronics. Copyright 2018
```

6.10 Programmable Parameters Information

DFL168A puts some parameters into flash memory. It will read these parameters into SRAM when power on or reset, and DFL168A will use these parameters to control default value of IC. These parameters can be changed by "PP xx SV hh" command at any time. "xx"(Hex) is parameter number, "hh"(Hex) is new value. For example, Parameter number 1 is "ATH" default value, 00 is "H1", FF is "H0". If we want to change the default value to H1, type the following commands:

```
>AT PP 01 SV 00
OK

> AT PP 01 ON
OK

>AT PPP
OK

>
```

The first command will change the value of parameter number 01 to 00, the second command will turn on parameter number 01, the third command will store all parameters into flash memory. Sometimes, you can ignore the third command if you execute some commands which contains the operation of writing flash memory. These commands are "AT SP h", "AT Z", "AT @R","ATD" and "AT DEV1 SP" . Of cause, DFL168A will store these parameters into flash memory before sleep.

You have to restart DFL168A if you want the change to happen.
Sometimes, you change many parameters, you want to see the current programmable parameters, you can use "AT PPS" to view them.

```
>AT PPS
00:00 F  01:FF F  02:00 N  03:32 F
04:00 F  05:00 F  06:33 F  07:F1 F
08:44 F  09:46 F  0A:4C F  0B:31 F
0C:36 F  0D:38 F  0E:41 F  0F:20 F
10:56 F  11:31 F  12:2E F  13:30 F
14:30 F  15:00 F  16:FF F  17:0D N
18:00 F  19:00 F  1A:FF F  1B:FF F
1C:57 F  1D:14 F  1E:F9 F  1F:FF F
20:FF F  21:FF F  22:FF F  23:FF F
24:FF F  25:FF F  26:FF F  27:96 F
28:FF F  29:34 F  2A:00 F  2B:00 F
2C:00 F  2D:00 F  2E:81 F  2F:00 F
30:00 F  31:00 F  32:02 F  33:E0 F
34:04 F  35:80 F  36:0A F  37:FF F
38:00 F  39:FF F  3A:FF F  3B:FF F
3C:FF F  3D:FF F

>
```

"N" denotes ON, "F" denotes OFF.

All Programmable Parameters are shown in the table below:

Programmable Parameters Table :

PP	Description	Values	Default
00	Reserved		

01	Display header bytes (ATH default value)	00=ON FF=OFF	FF (OFF)
02	Reserved		

03	"No Data" time out (AT ST default value)
	time = setting vale x 4ms but FB = 2 Sec
	FC=4 Sec FD=6 Sec FE=8 Sec FF=10 Sec
		00 to FF	32
			(200ms)

04	PD or VF output select PD: Power down VF: Vehicle Forward	00=PD Enable FF=VF Enable	00 (PD)
05	Vehicle Protocol on automatically when Power on /or Vehicle Protocol on when digital request	00 (On when request) FF (On when power on)	00
06	MID of tester for J1708	00 to FF	33
07	Reserved		
08 to 17	Reserved		
18	Reserved		
19	Character echo (ATE default value)	00=ON FF=OFF	00 (ON)
1A	PD or VF Active logic (AT PD or AT VF default value)	00 =VF0 or PD0 FF= VF1 or PD1	FF (AT PD1 or AT VF 1)
1B	Reserved		
1C	UART1 default Baud Rate setting Actual Baud Rate= 5000/setting value But actual Baudrate=9600 when setting is 00 Typical baud rate table	00 to FF	57 (57600)

Baud Rate bps	**PP 1C value Hex (Dec)**
9600	00 (0)
19200	FF (255)
38400	82 (130)
57600	57 (87)
115200	2B (43)

1D	Vehicle Speed out frequency setting: It is a frequency for each 1 km/h So Actual Frequency = setting x Vehicle Speed	0A to 28 (Hex) Or 10 to 40(Dec) You may use	14(Hex)

	(Km/h)	maximum of FF, but output frequency may be too high	
1E	J1939 Source Address	00 to FF	F9
1F to 28	Reserved		
29 to 30	64 bits' J1939 Name	00 to FF	34 00 00 00 00 81 00 00
31	User protocol's stuff byte for transmit	00 to FF	00
32	Protocol A, J1939 baud rate Actual Baud Rate= 500/Setting Value (Kbps)	01 to 40	02 (250Kbps)
33	USER 1 protocol (Protocol No: B) Setting: B7-----0: 29 bits ID transmitted 1: 11 bits ID transmitted B6-----0: Fixed 8 bytes data transmitted 1: non-fixed bytes data transmitted. B5-----0: ID received define by B7 1: Both 11 and 29 bits ID can be received. B4 ----Reserved B3------Reserved **B2 B1 B0** / **Format** 000 / none 001 / ISO15765 010 / SAE J1939		E0
34	USER 1 protocol (Protocol No: B) Baud rate Actual Baud Rate= 500K/setting value	01 to 40	04 (125K)
35	USER 2 protocol (Protocol No: C) Setting: The definition is the same as address 0x 33		80
36	USER 2 protocol (Protocol No: C) Baud rate Actual Baud Rate= 500K/setting value	01 to 40	0A (50K)

37	UART2 default Baud Rate setting Actual Baud Rate= 5000/setting value But Actual Baudrate=9600 when setting is 00 Actual Baudrate=4800 when setting is 01 Actual Baudrate=2400 when setting is 02 Typical baud rate table	00 to FF	FF (19200bps)

Baud Rate **bps**	**PP 1C value** **Hex (Dec)**
2400	02 (2)
4800	01 (1)
9600	00 (0)
19200	FF (255)
38400	82 (130)
57600	57 (87)
115200	2B (43)

38 to 3F	Reserved		

Notes: *1 Don't modify the parameters which is reserved, otherwise you will get unexpected results.*
2 If you modified parameter 1C, and you let the parameter on, then save it and restart IC, you find IC can't communicate with your hyperteminal. Only solution is that connect An0 to ground and restart IC, it will use the default baud rate 57600.

6.11 ERROR and Warning Information

DFL168A will Display the following information when error or warning occurs:

?

It means that DFL168A cannot understand the command user sends. In general, It is caused by type mistakes. Sometimes, it is caused by conditions of commands. For example, "AT DEV1 CSM hh" will display "?" if current device 1 is on. It means that you have to use "AT DEV1 PC" before executing "AT DEV1 CSM hh".

Address Claim fails because all arbitrary address failed

It means that J1939 address claim fails, and all arbitrary address failed

Address Claim fails because it cannot transmit address claim PGN

It means that J1939 address claim fails because transmitting claim failed

Address Claim fails and it is non-arbitrary device name
It means that J1939 address claim fails and it has non-arbitrary device name

Address Claim fails because two device names are the same
It means that there are the same device names in the J1939 network.

Can not transmit Command
It means that digit command for vehicle data bus transmit fails

CRC ERROR
It means that there is a CRC error when reading data from one-wire slave device

ECU Acknowledge
It is not an ERROR, it tells us that ECU responds with acknowledge to our request

ECU is busy, Try late
It notices us that ECU is busy, can't respond to the request right now

ECU Negative Acknowledge
It notices us that ECU responds with negative acknowledge to our request

ECU refuse request
It notices us that ECU refused our request.

Error: Can Bus Transmit
It means that we cannot transmit data in the CAN BUS. The reason perhaps is CAN BUS transceiver fault or no other devices connect CAN BUS network.

Error: My Address claim fails
It means that there is a general address error when DFL168A worked in J1939.

Error: NO TP.DT packet
It means that DFL168A expects J1939 TP.DT packet, but it can't get.

Error: The Same Device Name
It means that another node of J1939 with the my name occurred

Error: Too many Data packets
It means that receiving J1939 Data packets are more than expected

IC wakes up

It is a hint, not an error. It means that DFL168A wakes up

NODATA

It means that No data return for vehicle data request

No Device Connected

It means that DFL168A didn't find slave device on the one-wire bus.

No Response

It means that no data return for data request of Device 1.

Sleep

It is a hint or error. It means that DFL168A has slept.

Too many digit

It means that request command for vehicle data is too long.

Unable To Connect

It means that IC cannot find proper ISO15765 after searching all ISO15765 protocols

Wrong Digit Length

It means that the length of request command for vehicle data is wrong.

Warning Canceled!

It is a hint, not an error. It means that sleep warning has been canceled.

Warning: Sleep in hhhh Hex seconds

It is a hint, not an error. It means that DFL168A will sleep after hhhh hexadecimal seconds

7 Electrical Characteristics

Absolute Maximum Ratings

Ambient temperature under bias...-40°C to +125°C

Storage temperature .... -65°C to +160°C

Voltage on VDD with respect to VSS .... -0.3V to +4.0V

Voltage on any pin that is not 5V tolerant with respect to VSS ....-0.3V to (VDD + 0.3V)

Voltage on any 5V tolerant pin with respect to VSS when VDD = 3.0V............ -0.3V to +5.6V

Voltage on any 5V tolerant pin with respect to Vss when VDD < 3.0V................ -0.3V to 3.6V

Maximum current out of VSS pin ....300 mA

Maximum current into VDD pin..250 mA

Maximum current sourced/sunk by any 2x I/O pin..8 mA

Maximum current sourced/sunk by any 4x I/O pin..15 mA

Maximum current sourced/sunk by any 8x I/O pin..25 mA

Maximum current sunk by all ports ....200 mA

Maximum current sourced by all ports..200 mA

DC Characteristics

VDD normal operating range...3.0 to 3.6V

Analog Supply Voltage AVDD..Min: Greater of VDD – 0.3 or 3.0V

Analog Supply Voltage AVDD..Max: Lesser of VDD + 0.3 or 3.6V

Operating Junction Temperature Range TJ..-40 to +125°C

Operating Ambient Temperature Range TA..-40 to +85°C

Package Thermal Resistance θJA, 28-pin SPDIP.......................................45°C/W

Package Thermal Resistance θJA, 28-pin SOIC...50°C/W

Operating Current (Idd) when TA=-40°C ....Typical: 34mA

Operating Current (Idd) when TA=-40°C ....Max: 42mA

Operating Current (Idd) when TA=+25°C ....Typical: 34mA

Operating Current (Idd) when TA=+25°C ....Max: 41mA

Operating Current (Idd) when TA=+85°C ....Typical: 34mA

Operating Current (Idd) when TA=+85°C ....Max: 42mA

Sleep Current (TA=+25°C VDD=3.3V)..Typical:28uA

Sleep Current (TA=+25°C VDD=3.3V)..Max:87uA

Sleep Current when caused by AT command (TA=+25°C VDD=3.3V).................Typical:11.5mA

Sleep Current when caused by AT command (TA=+25°C VDD=3.3V)..................Max:13mA

Input Logic Low ViL ....Max: 0.2VDD

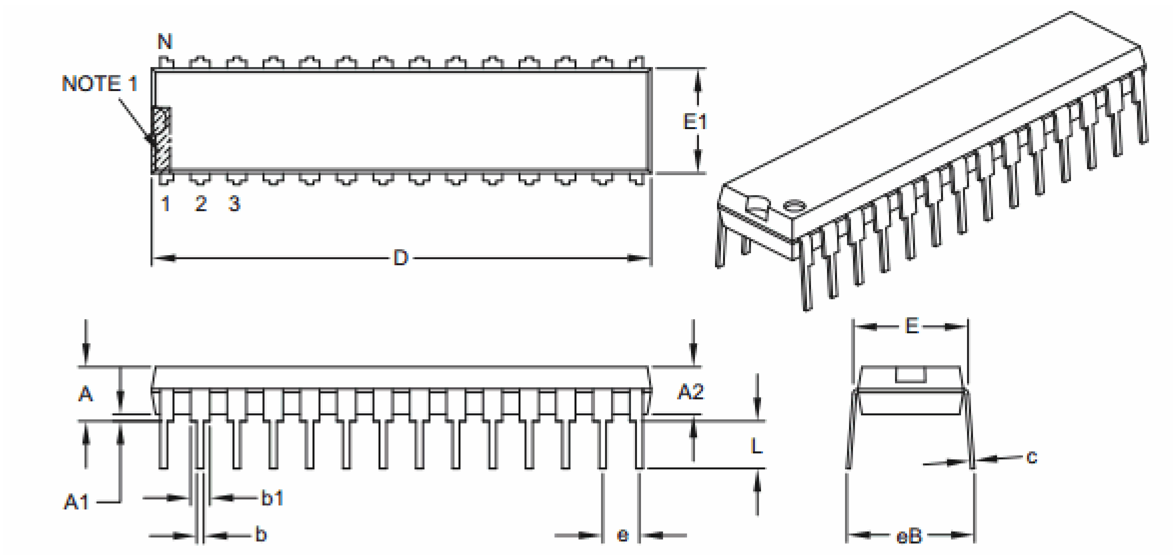
Input Logic High ViH..Min: 0.7VDD

Output Logic low VoL	Max:0.4V
Output Logic High VoH	Min:2.4V
Analog input voltage range	0 to AVDD

8 Packaging Diagrams and Parameters

Package Details

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]: Part number: DFL168A/P



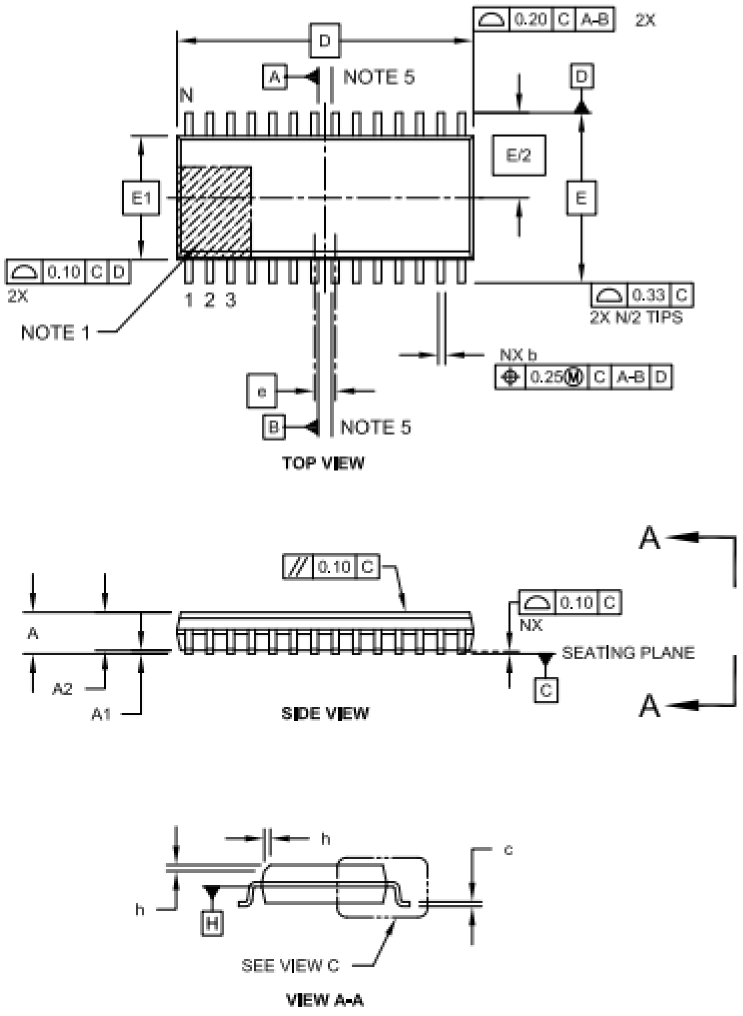
Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

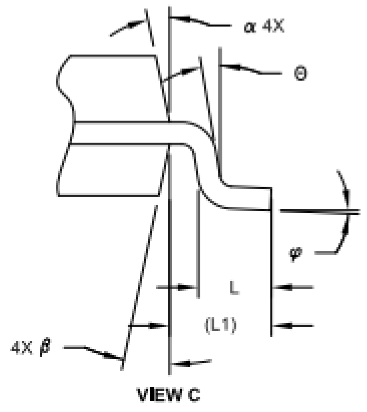
Notes:

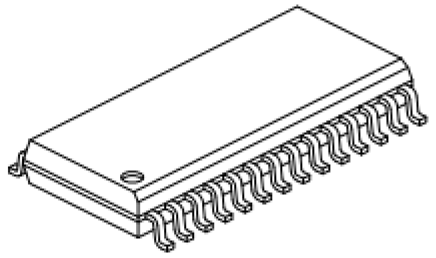
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

28-Lead Plastic Small Outline (SO) -Wide, 7.5mm Body [SOIC]: Part number: DFL168A/S





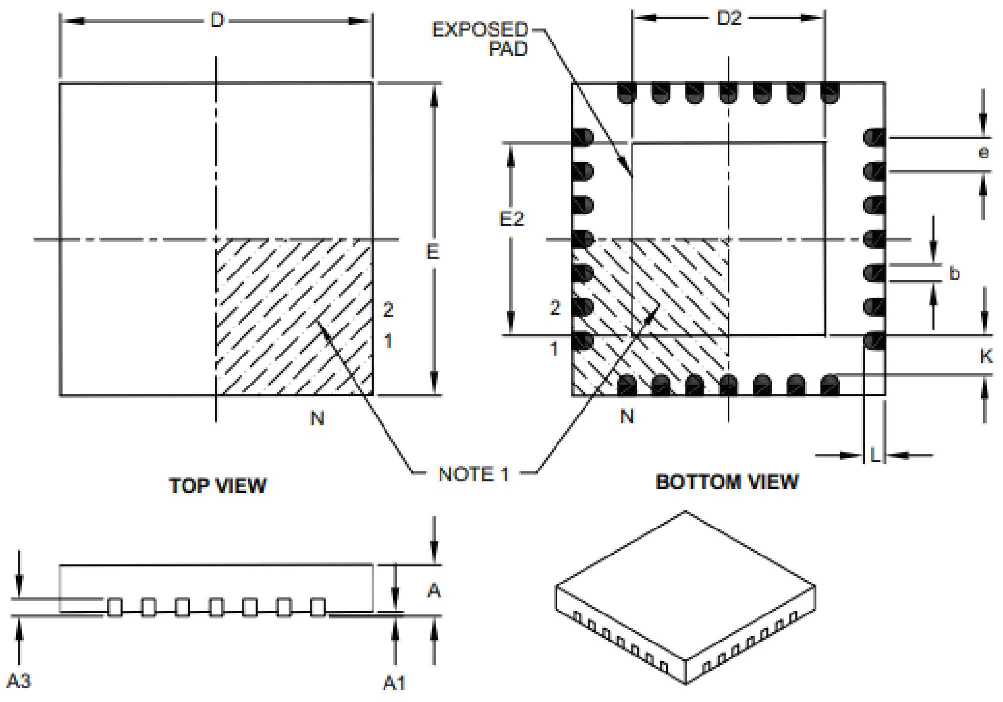


Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

28-Lead Plastic Quad Flat, No Lead Package (MM) – 6x6x0.9 mm Body [QFN-S] with 0.40 mm Contact Length Part number: DFL168A/S-QFN



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.70
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.70
Contact Width	b	0.23	0.38	0.43
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

IMPORTANT NOTICE

The information in this manual is subject to change without notice.

COPYRIGHT