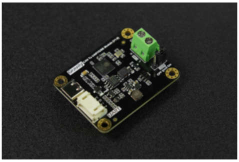
search
What is XX series
How to select series
How it works series
Development Boards
DFR0049 Analog Gas Sensor (QM-NG1)
DFR0100 DFRduino Beginner Kit For Arduino V3
DFR0133 X-Board
DFR0162 X-Board V2
DFR0188 Flymaple V1.1
DFR0182 Wirless GamePad V2.0
DFR0267 Bluno
DFR0282 Beetle
DFR0283 Dreamer Maple V1.0
DFR0296 Bluno Nano
DFR0302 MiniQ 2WD Plus
DFR0304 BLE Wireless Gamepad V2
DFR0305 RoMeo BLE
DFR0306 Bluno Mega 1280
DFR0321 Wido-WIFI IoT Node
DFR0323 Bluno Mega 2560
DFR0329 Bluno M3
DFR0339 Bluno Beetle
DFR0343 UHex Low-power

1. Introduction
2. Features
3. Application
4.Terminology Description
5. Functional diagram
6. Specification
7. Dimension
8.Initial use: AT command testing
9.Detailed explanation of AT command
10. CAN Communication Testing
11. More Documents
FAQ

SKU:TEL0150

1. Introduction



This is a CAN to UART and USB 1.0 communication module that is compatible with both 3.3V and 5V. It supports the CAN 2.0 protocol and incorporates the SLCAN command, eliminating the need for you to concern yourself with the complexities of the CAN protocol. By simply sending AT commands, you can engage in CAN data transmission and configure its functionalities.

This module not only provides a UART interface but also integrates a USB interface, eliminating the need for cumbersome conversions between USB and UART during data debugging. Connecting it to your computer via USB allows for convenient data debugging through a serial assistant. This feature proves particularly handy when deploying the module on-site for debugging purposes.

The module offers a configurable CAN baud rate range of 25K-1Mbps, supporting a maximum rate of 2 milliseconds per transmission and ensuring the delivery of 500 data packets per second without any loss. Its stable and reliable communication capabilities significantly reduce the likelihood of errors in your project and streamline the project debugging process.

Additionally, the module is equipped with data communication indicators and a 120Ω resistor jumper. These design details serve to enhance the efficiency of your project development and validation.

2. Features

- Supports CAN 2.0.
- Incorporates SLCAN protocol commands, ensuring user-friendly operation with no need to focus on CAN protocol intricacies.
- Enables CAN baud rate configuration ranging from 25K to 1Mbps, guaranteeing the delivery of 500 data packets per second without any loss.
- Integrates a USB interface, eliminating the requirement for additional USB-to-UART connections and providing convenient usability.
- Compatible with both 3.3V and 5V voltage levels and power supply.
- Configuration parameters can be saved even when power is disconnected.

3. Application

- Intended for communication among robotics and intelligent vehicles enthusiasts.

- Ideal for research and development engineers, as well as small-scale development companies, for learning, product testing, development, and integration purposes.
- Illumination control system for small vehicles.

4.Terminology Description

CAN

This is an ISO internationally standardized serial communication protocol.

The CAN network is comprised of CAN nodes and a CAN bus, possessing characteristics such as strong communication real-time performance, high reliability, multi-master operation, error detection, and fault confinement. Its bus protocol has become the standard for automotive computer control systems and embedded industrial control local area networks.

The CAN bus is a distributed control bus that utilizes differential voltage to transmit data, consisting of two signal lines: CAN_H and CAN_L. The bus has two logical levels: recessive level (logic 1) and dominant level (logic 0). When the voltage difference between CAN_H and CAN_L is less than 0.5V, it represents logic value 1, the recessive level. When the voltage difference between CAN_H and CAN_L is greater than 0.9V, it represents logic value 0, the dominant level. This differential transmission method enhances signal immunity to interference and suppresses interference from common-mode signals.

Taking the example of a high-speed, short-distance closed-loop network based on the ISO 11898 standard, the maximum communication distance of the CAN bus can reach up to 40m, with a maximum communication speed of 1Mbps. Additionally, a 120Ω termination resistor is required at the ends of the CAN bus to match the impedance and reduce echo reflections.

Data Frames

Communication on the CAN bus is conducted in the form of "frames." A data frame, as the name suggests, is a frame used to transmit data and serves as a carrier for the data that the transmitting node wishes to deliver to the receiving node.

The frame structure of a data frame consists of seven segments: Start of Frame, Arbitration, Control, Data, CRC, ACK, and End of Frame. Based on the length of the ID code in the arbitration segment, data frames are categorized into Standard Frames and Extended Frames. Standard Frames have an 11-bit binary ID, which translates to 3 hex digits, resulting in an ID range of 000-7FF. Extended Frames have a 29-bit ID, which translates to 8 hex digits, providing an ID range of 00000000-1FFFFFFF. The frame ID determines the priority of data frame transmission, with lower ID values indicating higher priority.

The Control segment consists of reserved bits r1 and r0, along with the Data Length Code (DLC) segment. The DLC segment is the most important, comprising 4 data bits used to indicate the number of bytes in the data segment of the message, ranging from 0 to 8.

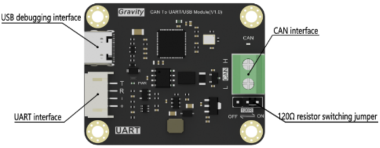
The Data segment consists of the transmitted data within the data frame, determined by the DLC, and each byte contains 8 bits.

The CRC segment includes the cyclic redundancy check sequence

(CRC) and the delimiter (DEL), with the former used to verify the correctness of the transmission and the latter indicating the end of the CRC sequence.

The ACK segment comprises the acknowledge (ACK) bit and the delimiter (DEL), confirming the successful reception.

5. Functional diagram


USB debugging interface
CAN interface
UART interface
120Ω resistor switching jumper
Gravity
CAN To UART/USB Module(V1.0)
CAN
H
CAN
L
T
R
-
+
PWR
120Ω
OFF
ON
UART

Functional name	**Pin name**	**Functional/pin description**
UART interface		**UART interface（Interface level3.3V、5V)**
	T	RX，Data reception port
	R	TX，Data transmission port
	-	Power negative terminal
	+	Power positive terminal （3.3V~5V)
CAN interface		**CANBUS interface**
	H	CAN_H，Can only be in a high-level or floating state
	L	CAN_L，Can only be in a low-level or floating state
120Ω resistor switching jumper		**Disconnect or connect 120Ω termination resistor**
	OFF	Disconnect 120Ω resistor
	ON	When the CAN device acts as a terminal device, the 120Ω resistor needs to be connected
USB debugging interface		**USB device interface, used for communication, testing, or parameter configuration with a computer**
CAN communication indicator light		LED flashes during CAN communication

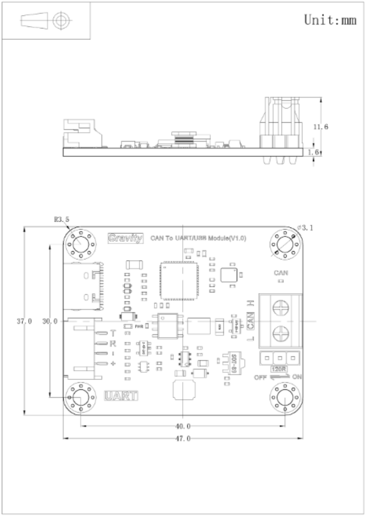
6. Specification

- Operating voltage: 3.3V - 5V
- Operating current: 65mA
- AT command protocol: SLCAN
- USB interface: USB serial device, requiring no communication

baud rate configuration.

- UART interface voltage levels: 3.3V, 5V
- UART baud rate: 115200Bps (default configuration)
- Optional baud rates for CAN communication:
  - 25kbits/s
  - 50kbits/s
  - 100kbits/s
  - 125kbits/s
  - 250kbits/s
  - 500kbits/s (default configuration)
  - 800kbits/s
  - 1Mbits/s

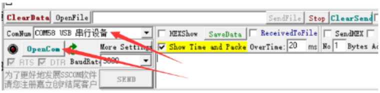
7. Dimension


Unit:mm
11.6
1.6
R3.5
⌀3.1
Gravity
CAN To UART/USB Module(V1.0)
CAN
H
CAN
L
37.0
30.0
T
R
-
+
PWR
UART
120R
OFF
ON
40.0
47.0

8.Initial use: AT command testing

The CAN to TTL module is designed with a USB interface, allowing you to directly connect it to your computer for command testing. No additional USB to UART converter is required. It functions as a USB device rather than a UART serial port, so you don't need to worry about the serial port baud rate. You can communicate at any baud rate of your choice (such as 9600 or 115200) with ease.

Connect the CAN to TTL module to your computer's USB port using a TYPE-C data cable. Once connected, open the serial assistant. You will see a USB serial device listed in the serial port options. Select that device and click on "OpenCom" to establish the serial connection, as shown in the diagram.


ClearData
OpenFile
SendFile
Stop
ClearSend
ComNum
COM58 USB 串行设备
OpenCom
More Settings
RTS
DTR
BaudRate
☐ HEXShow
SaveData
☐ ReceivedToFile
☐ SendHEX
☑ Show Time and Packe
OverTime: 20 ms
No 1 Bytes
为了更好地发展SSCOM软件
请您注册嘉立创P结尾客户
SEND

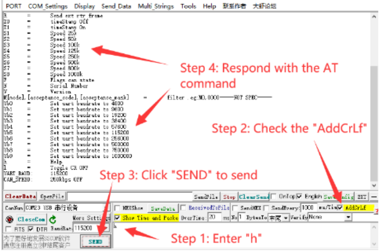
Now, let's try entering the command `h\r` to retrieve the help information. Here's how you can do it:

Step 1: Enter "h\r" or simply "h" (if "AddCrLf" is not selected, you need to enter "h\r") in the command area.

Step 2: If you only enter "h," make sure to check "AddCrLf."

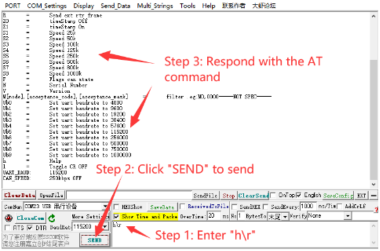
Step 3: Click on "SEND" to send the command "h."

Step 4: After sending, the data window will display the received help information, including all the commands and the current CAN baud rate details.



```
R          =        Send ext rtr frame
Z0         =        timeStamp Off
Z1         =        timeStamp On
S1         =        Speed 20k
S2         =        Speed 50k
S3         =        Speed 100k
S4         =        Speed 125k
S5         =        Speed 250k
S6         =        Speed 500k
S7         =        Speed 800k
S8         =        Speed 1000k
F          =        Flags can state
N          =        Serial Number
V          =        Version
M[mode],[acceptance_code],[acceptance_mask]      =        filter .eg:M0,0000——NOT SPEC——
Ub0        =        Set uart baudrate to 4800
Ub1        =        Set uart baudrate to 9600
Ub2        =        Set uart baudrate to 19200
Ub3        =        Set uart baudrate to 38400
Ub4        =        Set uart baudrate to 57600
Ub5        =        Set uart baudrate to 115200
Ub6        =        Set uart baudrate to 256000
Ub7        =        Set uart baudrate to 500000
Ub8        =        Set uart baudrate to 750000
Ub9        =        Set uart baudrate to 1000000
h          =        Help
l          =        Toggle CR OFF
UART_BAUD:          115200
CAN_SPEED:          250kbps OFF
```

PORT
COM_Settings
Display
Send_Data
Multi_Strings
Tools
Help
联系作者
大虾论坛
ClearData
OpenFile
SendFile
Stop
ClearSend
OnTop
English
SaveConfig
EXT
ComNum
COM23 USB 串行设备
CloseCom
More Settings
RTS
DTR
BaudRate
115200
HEXShow
SaveData
ReceivedToFile
SendHEX
SendEvery: 1000 ms/Tim
AddCrLf
Show Time and Packe
OverTime: 20 ms
No 1 BytesTo
末尾
Verify None
h
SEND
Step 4: Respond with the AT command
Step 2: Check the "AddCrLf"
Step 3: Click "SEND" to send
Step 1: Enter "h"



```
R          =        Send ext rtr frame
Z0         =        timeStamp Off
Z1         =        timeStamp On
S1         =        Speed 20k
S2         =        Speed 50k
S3         =        Speed 100k
S4         =        Speed 125k
S5         =        Speed 250k
S6         =        Speed 500k
S7         =        Speed 800k
S8         =        Speed 1000k
F          =        Flags can state
N          =        Serial Number
V          =        Version
M[mode],[acceptance_code],[acceptance_mask]      =        filter .eg:M0,0000——NOT SPEC——
Ub0        =        Set uart baudrate to 4800
Ub1        =        Set uart baudrate to 9600
Ub2        =        Set uart baudrate to 19200
Ub3        =        Set uart baudrate to 38400
Ub4        =        Set uart baudrate to 57600
Ub5        =        Set uart baudrate to 115200
Ub6        =        Set uart baudrate to 256000
Ub7        =        Set uart baudrate to 500000
Ub8        =        Set uart baudrate to 750000
Ub9        =        Set uart baudrate to 1000000
h          =        Help
l          =        Toggle CR OFF
UART_BAUD:          115200
CAN_SPEED:          250kbps OFF
```

PORT
COM_Settings
Display
Send_Data
Multi_Strings
Tools
Help
联系作者
大虾论坛
ClearData
OpenFile
SendFile
Stop
ClearSend
OnTop
English
SaveConfig
EXT
ComNum
COM23 USB 串行设备
CloseCom
More Settings
RTS
DTR
BaudRate
115200
HEXShow
SaveData
ReceivedToFile
SendHEX
SendEvery: 1000 ms/Tim
AddCrLf
Show Time and Packe
OverTime: 20 ms
No 1 BytesTo
末尾
Verify None
h\r
SEND
Step 3: Respond with the AT command
Step 2: Click "SEND" to send
Step 1: Enter "h\r"

9.Detailed explanation of AT command

This module adopts the SLCAN protocol, integrating it into the MCU. Serial commands can be used to transmit data or make corresponding configurations. Additionally, communication with a PC supporting the SLCAN protocol is also possible.

By utilizing the SLCAN protocol, you can spare yourself from the complexities of the underlying CAN protocol. Achieving CAN communication is as simple as sending serial commands. Below, we will delve into the purposes and usage of each instruction within the SLCAN protocol.

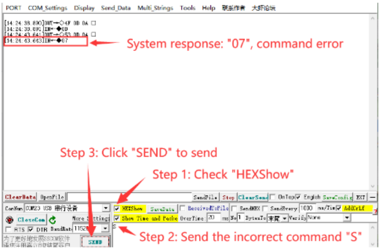
> Note: AT commands are case-sensitive.

9.1 Enable or disable CAN communication:

O = Activate CAN communication

C = Deactivate CAN communication

Command description: Use the commands "O" and "C" to enable or disable CAN communication, following the same method as before. If "HEXShow" is selected, the system will respond with 0D, indicating successful command execution. A response of 07 indicates an incorrect instruction.


PORT
COM_Settings
Display
Send_Data
Multi_Strings
Tools
Help
联系作者
大虾论坛
ClearData
OpenFile
SendFile
Stop
ClearSend
OnTop
English
SaveConfig
EXT
HEXShow
SaveData
ReceivedToFile
SendHEX
SendEvery
AddCrLf
CloseCom
More Settings
Show Time and Packe
OverTime
SEND

```
[14:24:38.890]OUT→◇4F 0D 0A □
[14:24:38.891]IN←◆0D
[14:24:43.641]OUT→◇53 0D 0A □
[14:24:43.643]IN←◆07
```

System response: "07", command error

Step 3: Click "SEND" to send

Step 1: Check "HEXShow"

Step 2: Send the incorrect command "S"

9.2 Sending CAN data:

t = Send 11-bit standard data frame

r = Send 11-bit standard RTR data frame

T = Send 29-bit extended data frame

R = Send 29-bit extended RTR data frame

Z0 = Data frame without timestamp

Z1 = Data frame with timestamp

Command description:

The SLCAN protocol allows for the transmission of 11-bit standard data frames and 29-bit extended data frames. The only difference between CAN standard data frames and extended data frames is the length of the frame ID, enabling the connection of more CAN nodes.

When sending data, adhere to the frame length specifications in the SLCAN protocol. The ASCII representation of an SLCAN frame is as follows:

type: Indicates the type of data frame to be sent.

id: For standard data frames, the ID is represented by 3 bytes of ASCII Hex. For extended data frames, it is represented by 8 bytes of ASCII Hex.

dlc: A single-byte number (0-8) that describes the number of hexadecimal data to be sent.

data: The data to be sent in ASCII representation of hexadecimal, with a quantity corresponding to dlc.

Example:

Send 11-bit standard data frame: t123178 - CAN ID: 0x123, CAN DLC: 1, Data: 0x78

Send 11-bit standard data frame: t4563112233 - CAN ID: 0x456, CAN DLC: 3, Data: 0x11 0x22 0x33

Send 29-bit extended data frame: T12ABCDEF2AA55 - Extended CAN ID: 0x12ABCDEF, CAN DLC: 2, Data: 0xAA 0x55

Send 11-bit standard RTR data frame: r1230 - CAN ID: 0x123, CAN DLC: 0, No data, Remote Transmission Request.

9.3 CAN Communication Rate Setting:

S1 = Set CAN communication rate to 25 kbits/s

S2 = Set CAN communication rate to 50 kbits/s

S3 = Set CAN communication rate to 100 kbits/s

S4 = Set CAN communication rate to 125 kbits/s

S5 = Set CAN communication rate to 250 kbits/s

S6 = Set CAN communication rate to 500 kbits/s (default configuration)

S7 = Set CAN communication rate to 800 kbits/s

S8 = Set CAN communication rate to 1 Mbits/s

Command description:

The S1 to S8 commands are used to set the baud rate for CAN communication. In a CAN bus, all nodes must have consistent baud rates to communicate with each other.

Using the "h" command, you can query the current module's baud rate and the status of CAN communication being open or closed. Note that this sets the communication baud rate for the CAN bus, not the UART baud rate. The default baud rate for the UART interface is 115200.

9.4 Information Retrieval:

V = Get current version

h = Get help information

Command description:

Send "V" to retrieve firmware version information. After a successful transmission, the response will contain the version number. Send "h" to get help information, which will provide instructions for all commands as well as the current CAN baud rate and status.

10. CAN Communication Testing

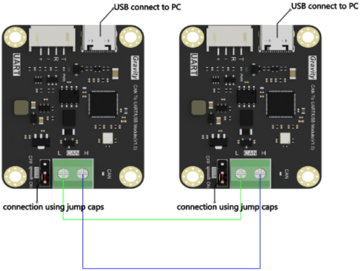
10.1 Communication between two modules.

This example will demonstrate the communication between two CAN-to-TTL modules. One module will send the data "t123122," while the other module will receive the data and display it on the serial monitor as "t123122."

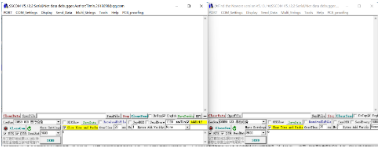
Hardware Preparation:

- Two CAN-to-TTL modules ×2
- Type-C USB cables ×2

Connection Diagram:


USB connect to PC
USB connect to PC
UART
Gravity
CAN
L
H
connection using jump caps
connection using jump caps

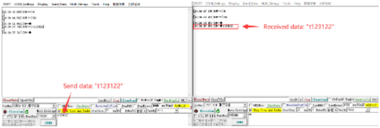
Open two serial assistants and select the corresponding serial devices for the two CAN-to-TTL modules, as shown in the diagram.



In both serial assistants, send the command "O" to activate CAN communication, followed by the command "S6" to set the CAN baud rate to 500 kbits/s for both modules.

In the left serial assistant, send the command "t123122."

In the right serial assistant window, the data "t123122" will be received.


Received data: "t123122"
Send data: "t123122"

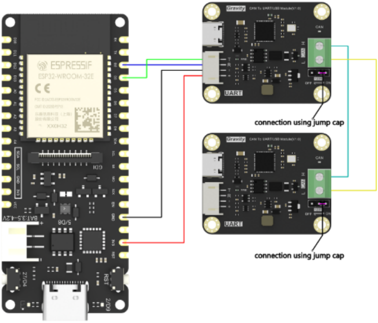
10.2 Tutorial for Using FireBeetle 2 ESP32-E as the Main Controller

This tutorial will demonstrate how to send data "t11123333" to the CAN bus through the UART interface of ESP32-E. Another CAN-to-TTL module will receive the data and display it on the serial monitor.

Hardware Preparation:

- CAN TO UART/USB Module ×2
- FireBeetle_Board_ESP32_E (SKU:DFR0654) ×1
- Type-C&Micro 2-in-1 USB Cable （SKU： FIT0668） ×2

Connection Diagram:


ESPRESSIF
ESP32-WROOM-32E
Gravity
UART
connection using jump cap
Gravity
UART
connection using jump cap

Example program

```
void setup()
{
  Serial1.begin(115200, SERIAL_8N1, /*rx =*/D3, /*tx =*/D2);
  delay(10); //Give a delay to ensure that the previous setup
  Serial1.println("S3\r"); //Set the CAN baud rate to 100kbits
   //delay(10); //Give a delay to ensure that the previous set
  Serial1.println("O\r"); //Turn on CAN communication
}

void loop()
{
  Serial1.println("t11123333\r");
  delay(1000);
}
```

Operational Steps:

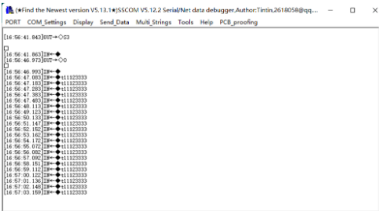
Connect the module to the FireBeetle Board ESP32-E following the wiring diagram shown above.

Open the Arduino IDE and upload the code to the FireBeetle Board ESP32-E.

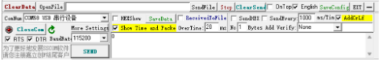
Open another serial assistant and select the corresponding serial device for the module. In the serial assistant, send "S3" to set the CAN baud rate to 100 kbits/s. Then, send the command "O" to activate CAN communication. You will then be able to receive the data "t11123333" in the serial assistant.

Note: The baud rate of the module's UART interface needs to match the baud rate selected in the serial assistant.

Demonstration of the Running Results


(★Find the Newest version V5.13.1★)SSCOM V5.12.2 Serial/Net data debugger,Author:Tintin,2618058@qq....
PORT
COM_Settings
Display
Send_Data
Multi_Strings
Tools
Help
PCB_proofing

```
[16:56:41.843]OUT→◇S3
□
[16:56:41.863]IN←◆
[16:56:46.973]OUT→◇O
□
[16:56:46.993]IN←◆
[16:56:47.083]IN←◆t11123333
[16:56:47.183]IN←◆t11123333
[16:56:47.283]IN←◆t11123333
[16:56:47.383]IN←◆t11123333
[16:56:47.483]IN←◆t11123333
[16:56:48.113]IN←◆t11123333
[16:56:49.123]IN←◆t11123333
[16:56:50.133]IN←◆t11123333
[16:56:51.147]IN←◆t11123333
[16:56:52.152]IN←◆t11123333
[16:56:53.162]IN←◆t11123333
[16:56:54.172]IN←◆t11123333
[16:56:55.072]IN←◆t11123333
[16:56:56.082]IN←◆t11123333
[16:56:57.092]IN←◆t11123333
[16:56:58.151]IN←◆t11123333
[16:56:59.112]IN←◆t11123333
[16:57:00.122]IN←◆t11123333
[16:57:01.136]IN←◆t11123333
[16:57:02.148]IN←◆t11123333
[16:57:03.159]IN←◆t11123333
```



ComNum: COM58 USB 串行设备

BaudRate: 115200

☑ RTS ☑ DTR

☐ HEXShow ☐ ReceivedToFile ☐ SendHEX ☐ SendEvery: 1000 ms/Tim ☑ AddCrLf

☑ Show Time and Packe OverTime: 20 ms No 1 Bytes Add Verify: None

☐ OnTop ☑ English

ClearData
OpenFile
SendFile
Stop
ClearSend
SaveConfig
EXT
CloseCom
More Settings
SaveData
SEND
0
为了更好地发展SSCOM软件
请您注册嘉立创P结尾客户

11. More Documents

TEL0150_尺寸图.pdf

TEL0150_2D_CAD文件.rar

FAQ

No customers have raised any inquiries regarding this product yet. Feel free to contact us through QQ or the forum!

For more questions and interesting applications, you can **visit the forum** to browse or post.

Get **CAN to TTL Communication Module with SLCAN Protocol** from DFRobot Store or **DFRobot Distributor**.

Turn to the Top

Downloaded from Arrow.com.