JCOM.CAN.BTS CAN Bus, OBD-II Bluetooth Scanner

Device Setup

https://copperhilltech.com

Table of Content

Introduction

The JCOM.CAN.BTS is a CAN Bus and OBD-II scanner device with Bluetooth connection. It scans user-defined CAN message frames and transmits them per Bluetooth to a PC or Android phone or tablet (A BLE - Bluetooth Low Energy - version is in planning to support communication with iOS devices as well). The hardware is based on a generic design that allows a multitude of applications with the support of optional electronic components and corresponding firmware.

The input power (7 VDC to 36 VDC to suit a great range of applications, including vehicles) and the CAN Bus / OBD-II connections are provided per DSUB-9 connector according to CiA recommendations. An additional RS232 port is used for setting up the scanner, e.g., CAN Bus baud rate, message ID length, and up to 100 message filters. The CAN Bus message frame is translated into an ASCCI string for easy readability and processing.

Features

- ARM Cortex-M3 Processor
- CAN Bus Interface
- Fully certified low-power Bluetooth Version 2.1 Module (with or without external antenna)
- RS232 Interface For On-Site Firmware Upload and/or Setup Functionality
- SD Card Slot for Setup Data Storage
- Extended Temperature Range of -40C to +85C (-25C to +85C with SD Card)
- Input Power Range of 7 VDC to 36 VDC
- Flame Retardant ABS Enclosure 4.25 x 3.00 x 1.38 in / 107.95 x 76.20 x 35.05 mm
- Environmentally friendly, RoHS compliant

Specifications

- **CAN Interface**
  - CAN Controller integrated in microcontroller
  - Fully ISO 11898-compliant
  - Supports CAN 2.0A And CAN 2.0B
  - Bit rate up to 1 Mbit/s
- **Bluetooth**
  - Fully certified Bluetooth version 2.1
  - Backwards compatible with Bluetooth version 2.0, 1.2, and 1.1
  - Low power: 26 µA sleep, 3 mA connected, 30 mA transmit
  - Bluetooth SIG certified
  - Certifications: FCC, IC, CE
  - Two antenna options available: PCB trace and external antenna
  - Delivers up to 3 Mbps data rate for distances up to 20 meters (~60 feet)
  - Optional upgrade to support distances up to 100 meters (~300 feet)

Functionality

The JCOM.CAN.BTS can be connected and adjusted to any CAN Bus network.
The device setup allows the setting of CAN baud rates (CiA recommendations) and the message ID length (11/29-bit or both). The CAN message frames are translated into an ASCII string and then transmitted per Bluetooth.

Message Filters

The device setup also supports the setup of up to 100 CAN message ID filters in order to reduce the data stream transmitted per Bluetooth. Per default, the device will pass all message IDs. Message filtering is strongly recommended because:

1. We are using an industrial-strength Bluetooth transmitter that provides a transmission range of up to 300 feet (100 m), which cannot be accomplished by other commercial solutions. However, the serial baud rate is limited to 115,200 baud, which may cause problems at high CAN baud rates in combination with a high busload.
2. Focussing on CAN messages that matter will improve the performance of the application, especially when operated using an Android hardware.

All settings, once they have been set up, are stored permanently on the onboard SD card.

Data Conversion Protocol

The Data Conversion Protocol describes the translation of CAN message frames into serial data as transmitted by the Bluetooth port. We chose the ASCII protocol for ease-of-use on the Bluetooth receiving side (e.g. an Android tablet/phone).

Serial Connection

115,200 baud

No Parity

8 Data Bits

1 Stop Bit

Data Frame Architecture

The received CAN data frame is converted into a simple ASCII String (of constant length) as follows:

@	Message frame start
8 characters	Message ID (hex)
1 character	S=Standard 11-bit message ID; X=Extended 29-bit message ID
1 character	CAN data length (0…8)

16 characters	CAN data (hex)
2 characters	Checksum

Sample:
@0C10A040X81122334455667788AF

- **Message ID**: Regardless of the message ID length (11 or 29 bits), the message ID has a constant length of 4 bytes (8 ASCII characters). In case of a standard message (11-bit message ID) the leading four bytes will be zero.
- **CAN data**: The CAN data field has a constant length of 8 bytes. The field is filled with data according to the transmitted data length; all remaining data bytes are set to zero.
- **Checksum**: The checksum is a 2's complement checksum that is applied to all ASCII characters starting after the frame start character and ending before the first checksum character.

What is a 2's Complement Checksum?

A checksum is a technique to check data for transmission errors or tampering. If the last few bytes are the sum of all the preceding bytes, then any errors are likely to be detected.

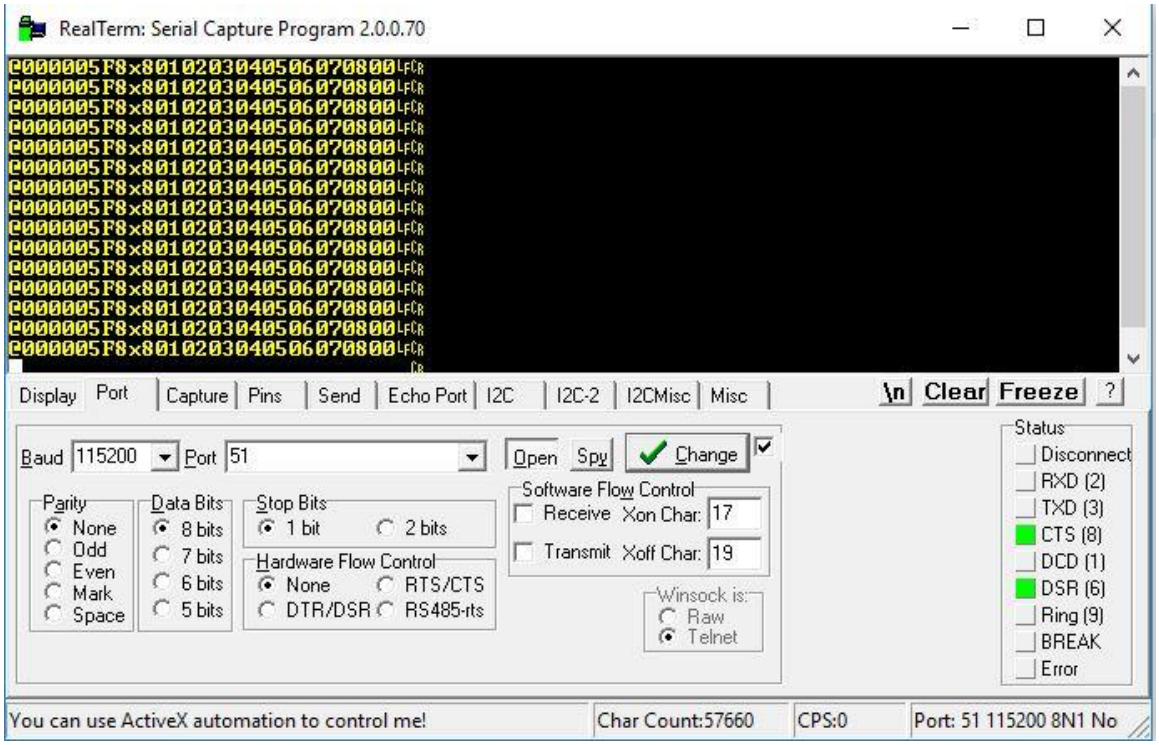
In this case, the checksum is the sum of all ASCII characters starting after the frame start character and ending before the first checksum character, ignoring any carry, meaning the checksum (hex) is always 8 bits long (0 to 255).

A one's complement is just a complement: ~x
A two's complement is a complement and increment, ignoring carry: (~x)+1

Bluetooth Data Display Per Terminal Program

After connecting and pairing the Bluetooth port to a Windows PC, you can start a terminal program with the above documented baud rate settings (115200, N, 8, 1) and select the COM port associated with the Bluetooth connection.

Note: Popular terminal programs for Windows are RealTerm and Tera Term.



When connected to a running CAN Bus network, you should see a data stream similar to the one displayed here. Please be aware that you need to synchronize thee CAN Bus baud rate setting between our gateway and the CAN network.

Device Setup

The JCOM.CAN.BTS is a CAN Bus and OBD-II scanner device with Bluetooth connection. It scans user-defined CAN message frames and transmits them per Bluetooth to a PC or Android phone or tablet.

Connecting Bluetooth to a Windows PC

- How to connect and pair a Bluetooth device with Windows 10...

Connecting Bluetooth to an Android Device

- Android Basics: How to connect a Bluetooth Device...

Device Setup

The RS232 interface is used for the device setup per PC (terminal program) and on-site firmware upload, using an RS232 to USB cable. Connect the device to your PC (which can be either Windows or Linux) and start a terminal program and select to the corresponding COM port. The device setup is managed at a serial baud rate of 9600 baud, which is the default for the majority of terminal programs.

Note: Popular terminal programs for Windows are RealTerm and Tera Term.

At startup and when hitting the 'h' key, the following help screen will appear:

```
HELP Menu:
a - [a]dd Message Filter
b - Set CAN [b]audrate
c - Display [c]urrent CAN Settings
d - [d]elete Message Filter
h - Print This [h]elp Menu
f - [f]LASH programming
i - Set Message [i]D length (11/29 bit)
m - Show Active [m]essage Filters
p - Toggle [p]ass All CAN Messages
r - [r]eset All Message Filters
s - [s]tore Parameters
```

a - Add Message Filter

Hit the 'a' key to add a message filter, and the following information will appear:

```
a
Add Message ID (hex):
Hit <Enter> to submit or 'x' to cancel...

00000000 added.
```

In this example, we added the message ID 00000000.

Note: Please be aware that the gateway will allow all message IDs per default. As soon as you apply one or more message filters, only those message IDs will be allowed.

b - Set CAN Baud Rate

Hit the 'b' key to set the CAN baud rate. The following screen will appear:

```
b
CAN Baudrate Settings:
1 - 5k
2 - 10k
3 - 20k
4 - 40k
5 - 50k
6 - 80k
7 - 100k
8 - 125k
9 - 200k
a - 250k
b - 500k
c - 1000k
Hit any other key to leave this menu...
CAN Baudrate = 500k
```

In this example, we hit 'b' to select a CAN baud rate of 500 kbit/sec.

c - Display Current CAN Settings

Hit 'c' to display all current CAN settings, such as CAN Bus baud rate and selected message ID length.

```
c
Current CAN Settings:
CAN Baudrate = 500k
Message ID = Both 11 & 29-bit
```

d - Delete Message Filter

Hit the 'd' key to delete a message ID filter.

```
d
Delete Message ID (hex):
Hit <Enter> to submit or 'x' to cancel...
1f00
00001F00 deleted.
```

In this example, we deleted the filter for message ID 1F00. The response will be as displayed in the image, or the system will point out that the ID was not part of the message filter list.

f - Flash Programming

Hit the 'f' key to initiate the flash programming feature, and confirm with Y/N.

```
f
Switch to Flash programming mode? (Y/N)
Flash programming mode initiated. Please close terminal and proceed with FlashMagic.
```

i - Set Message ID Length

Hit the 'i' key to set the message ID length.

```
i
CAN Message ID Length
s - Standard 11-bit ID
x - Extended 29-bit ID
b - Both
Hit any other key to leave this menu...
Message ID = Extended 29-bit
```

Per default, the gateway will read and forward CAN Bus message frames with either an 11- or 29-bit message ID. This feature allows the user to limit the number of allowed message frames to those that match the selected message ID length.

m - Show Active Message Filters

Hit the 'm' key to display all active CAN Bus message ID filters.

```
m
Active Message Filters:
000005F0
000005F1
000005F2
000005FF
```

p - Toggle - Pass All Message Filters

Hit the 'p' key to pass all message IDs for temporary use and hit 'p' again to turn the feature off and activate the currently active message filters.

```
p
Pass All - ON
p
Pass All - OFF
```

r - Reset All Message Filters

Hit 'r' to delete all CAN Bus message filter IDs and confirm with Y/N.

```
r
Reset all message filters? (Y/N)
All message filters have been cleared.
```

s - Store Parameters

Hit 's' to store all CAN Bus settings (CAN baud rate, message ID length, and message ID filters) to the internal SD card.

```
s
Parameters have been saved permanently.
```