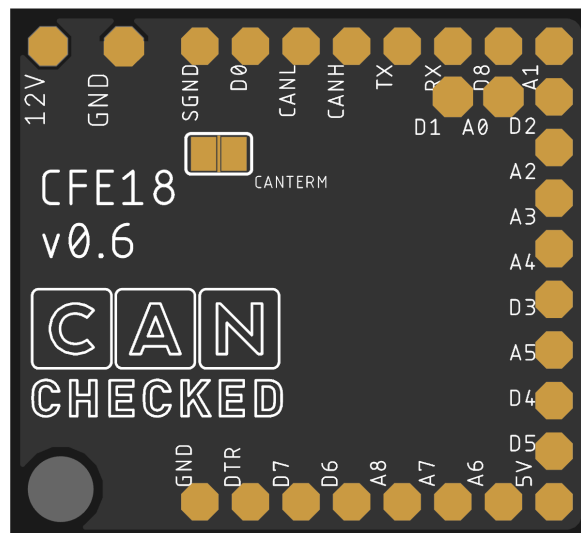


# CFE

## Can Bus Function Extender

### Anleitung/Instruction



(not for open road usage!)

## 1. Einführung

Über Can Bus lassen sich verschiedenste Daten übertragen. Steuergerät können flexibel Daten empfangen, wodurch sich die Funktionalität mit weiteren Sensoren erweitern lässt.

Möchte man zum Beispiel Dreh- und Druckknöpfe an seinem Lenkrad anbringen, benötigt man eine sehr kompakte Can Bus Lösung.

## 2. Lösung

CANchecked bietet ein kleines Zusatzmodul, welches 18 Erweiterungsmöglichkeiten bietet – 9 analoge (0-5V) und 9 digitale (5V tolerant).

## 3. Hardware

Das Board ist mit nur 27x30mm möglichst kompakt gehalten. Die Pins haben einen Abstand von 2.54mm. Nur 12V und Masse für die Versorgung sind aus Sicherheitsgründen weiter auseinander (3,81mm).

Die KFZ Boardspannung wird von maximal 15V auf 5V gewandelt, welche als Versorgungsspannung für den Microcontroller als auch für externe Sensorik nutzbar ist.

Der Stromverbrauch liegt bei 12V/15mA.

## 4. Optionen

Auf der Rückseite befindet sich ein Lötjumper. Wird dieser gebrückt, so aktiviert man die 120 Ohm Terminierung für den Can Bus.

Über Can Bus lassen sich folgende Werte anpassen:

## 1. Introduction

A wide variety of data can be transmitted via Can Bus. ECUs units can receive data from Can Bus, which means they can expand their functionality with additional sensors.

For example, if you want to attach knobs and push buttons to your steering wheel, you need a very compact Can Bus solution.

## 2. Solution

CANchecked now offers a small additional module that offers 18 expansion options - 9 analog (0-5V) and 9 digital (5V tolerant).

## 3. Hardware Add-on

The board is kept as compact as possible with only 27x30mm. The pins are 2.54mm apart. Only 12V and ground for the power supply are further apart for safety reasons (3.81mm).

The vehicle board voltage is converted from a maximum of 15V to 5V, which is used as a supply voltage for the microcontroller and can also be used for external sensors.

The power consumption is 12V / 15mA.

## 4. Options

There is a solder jumper on the back. If this is bridged, the 120 Ohm termination for the Can Bus is activated.

The following values can be adjusted via Can Bus:

- |   |  |
|---|--|
| <ol style="list-style-type: none"> <li>1) <b>Data CAN ID:</b> mit welcher Can Id die Daten gesendet werden –default: 0x700</li> <li>2) <b>Config CAN ID:</b> mit welcher CAN Id das Board konfiguriert werden kann (default: 0x70A) – NEUSTART erforderlich</li> <li>3) <b>Frequenz</b> mit der die Daten über dan Can Bus geschickt werden – maximal 15ms=66Hz (Default: 50ms = 20Hz)</li> <li>4) <b>Can Bus Speed:</b> 1=125kbps, 2=250kbps,3=500kbps,4=1Mbit (Default: 3 = 500kbps) – NEUSTART erforderlich</li> <li>5) <b>Modus:</b> 0=default, 1=EMUv3, 2=Haltech IO12B, 3=Haltech IO12A+B, 4=Motec</li> </ol> | <ol style="list-style-type: none"> <li>1) <b>Data CAN ID:</b> decide which Can id is used – default: 0x700</li> <li>2) <b>Config CAN ID:</b> decide which Can id can be configured (default: 0x70A) – restart needed</li> <li>3) <b>Frequency</b> with which tha data is being send via Can Bus up to 15ms which is 66Hz (Default: 50ms = 20Hz)</li> <li>4) <b>Can Bus Speed:</b> 1=125kbps, 2=250kbps,3=500kbps,4=1Mbit (Default: 3 = 500kbps) – restart needed</li> <li>5) <b>Mode:</b> 0=default, 1=EMUv3, 2=Haltech IO12B, 3=Haltech IO12A+B, 4=Motec</li> </ol> |
|---|--|

Byte0+1 are always 0x0C and 0x0A

Byte2 gives the feature starting with 0x0A  
 can id: 0x0A (high byte) + 0x0B (lowbyte)  
 config can id: 0x0C (high byte) + 0x0D (lowbyte)  
 frequency: 0x0E  
 can bus speed: 0x0F  
 mode: 0x10

**Example 1:** change data can id to 0x600

0x70A 0x0C 0x0A 0x0A 0x06

0x70A 0x0C 0x0A 0x0B 0x00

**Example 2:** change frequency to 50Hz (20ms)

0x70A 0x0C 0x0A 0x0E 0x14

**Example 3:** change can bus speed to 1Mbit

0x70A 0x0C 0x0A 0x0F 0x4

**Example 4:** change mode to 1

0x70A 0x0C 0x0A 0x10 0x1

## 5. Can Bus Daten (default)

Die Daten der Eingänge werden abgefragt und alle 50ms übertragen (20Hz - änderbar). Die Daten werden als „unsigned big endian“ übertragen.

## 6. Can Bus Data (default)

The data of the inputs are queried and transmitted every 50ms (20Hz - changeable). The data is transmitted as "unsigned big endian".

Can Bus ID: 0x700  
(Data CAN ID – changeable)  
Byte0+1: AIN0 0-1023  
Byte2+3: AIN1 0-1023  
Byte4+5: AIN2 0-1023  
Byte6+7: AIN3 0-1023

Can Bus ID: Data CAN ID +1  
Byte0+1: AIN4 0-1023  
Byte2+3: AIN5 0-1023  
Byte4+5: AIN6 0-1023  
Byte6+7: AIN7 0-1023

Can Bus ID: Data CAN ID +2  
Byte0+1: AIN8 0-1023  
Byte2: bit masked DIN0-7  
Byte3: DIN0 0-1  
Byte4: DIN1 0-1  
Byte5: DIN2 0-1  
Byte6: DIN3 0-1  
Byte7: DIN4 0-1

Can Bus ID: Data CAN ID +3  
Byte0: DIN5 0-1  
Byte1: DIN6 0-1  
Byte2: DIN7 0-1  
Byte3: DIN8 0-1

## 7. Anschluss

12V und Masse verbinden. Die grüne Status LED auf der Vorderseite leuchtet.

Can High und Low mit dem Fahrzeug-Can Bus verbinden.

5V und SGND nutzt man für die analogen Eingänge.

## 7. Connection

Connect 12V and ground. The green status LED on the front lights up.

Connect Can High and Low with the vehicle Can Bus.

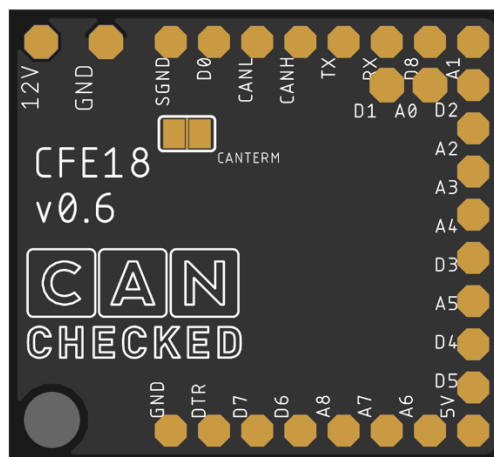
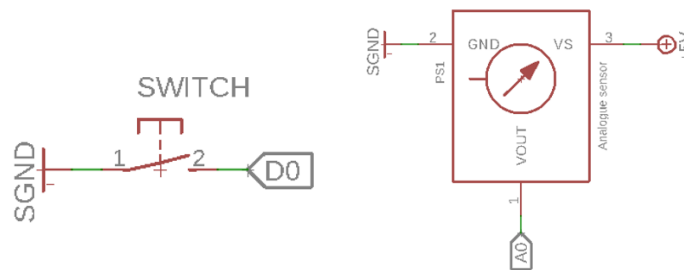
5V and SGND are used for the analog inputs.

Möchte man einen Taster oder Schalter nutzen, schaltet man diesen gegen SGND. Intern sind die digitalen Eingänge D0-D8 mit einem Pullup versehen.

Analoge Sensoren verbindet man mit 5V, SGND und einem analogen Eingang z.B. A0.

If you want to use a button or switch, you switch it against SGND. Internally, the digital inputs D0-D8 are provided with a pull-up.

Analogue sensors are connected to 5V, SGND and an analog input e.g. A0.



## 8. Steuergerät Konfiguration

Die analogen Eingänge liefern 10bit Auflösung und somit Werte von 0 bis 1023. Im Steuergerät muss der Wert umgerechnet werden. Alle Werte sind „unsigned big endian“

Beispiele:

**5V Spannung AIN0**

0x700 Byte0

Multiplikator: 5; Divisor: 1023

oder Multiplikator: 0.004887585533

**Abgastemperatur (type k) AIN7:**

0x702 Byte6

Multiplikator: 1250; Divisor: 1023

## 9. ECU config

The analog inputs provide 10-bit resolution and thus values from 0 to 1023. The value needs to be converted in the ECU to the correct value. All values are unsigned big endian.

Examples:

**5V voltage AIN0**

0x700 byte0

Muliplier: 5; divisor: 1023

Or multiplier: 0.004887585533

**Exhaust gas temperature (type k) AIN7:**

0x702 byte6

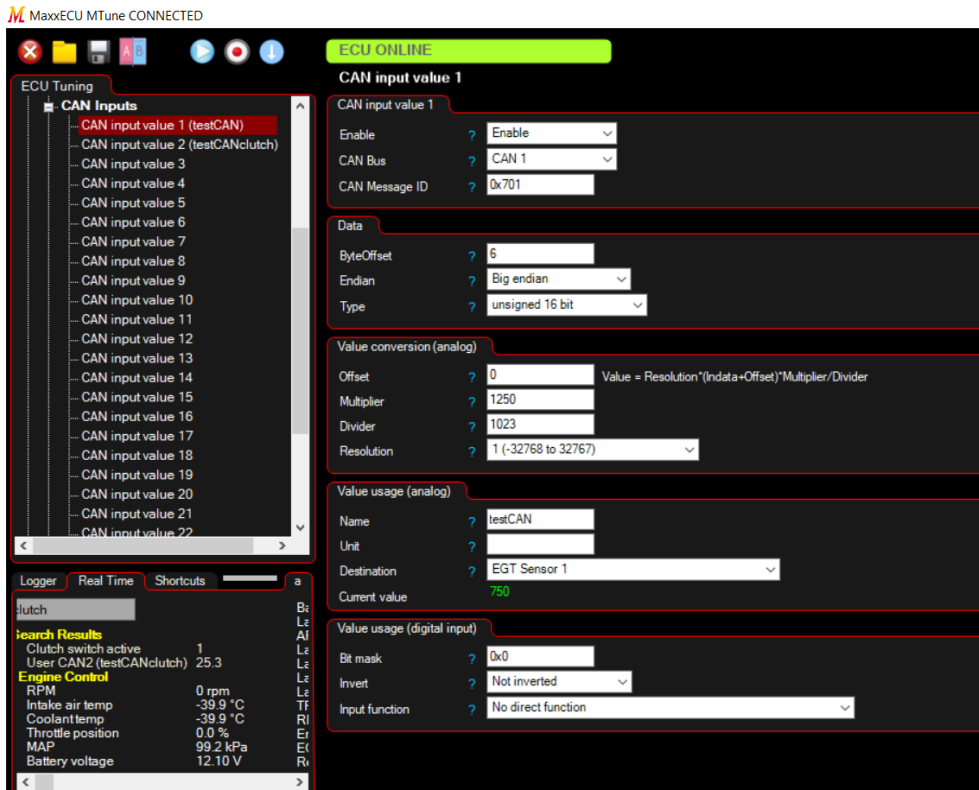
Muliplier: 1250; divisor: 1023

oder Multiplikator: 1.2218963832

Für die digitalen Eingänge nutzt man entweder die einzelnen Bytes. Z.B. DIN6: 0x702 Byte6  
oder die Bitmask bei 0x702 Byte2  
Z.B. DIN6:  
0x702 Byte2 Mask:0x40

Or multiplier: 1.2218963832

For the digital inputs you either use the bytes – e.g. DIN6:  
0x702 byte6  
or bitmask 0x702 byte2  
e.g. DIN6:  
0x702 byte2 mask:0x40



The screenshot displays the MaxxECU MTune software interface. At the top, a green bar indicates 'ECU ONLINE'. The main window is divided into several sections:

- ECU Tuning - CAN Inputs:** A list of 22 CAN input values. 'CAN input value 2 (testCANclutch)' is highlighted in red.
- CAN input value 2 configuration:**
  - Enable:
  - CAN Bus:
  - CAN Message ID:
  - Data:
    - ByteOffset:
    - Endian:
    - Type:
  - Value conversion (analog):
    - Offset:  Value = Resolution\*(InData+Offset)\*Multiplier/Divider
    - Multiplier:
    - Divider:
    - Resolution:
  - Value usage (analog):
    - Name:
    - Unit:
    - Destination:
    - Current value:
  - Value usage (digital input):
    - Bit mask:
    - Invert:
    - Input function:
- Logger - Real Time - Shortcuts:** A table of real-time data:

Parameter	Value	Unit
Clutch switch active	1	
User CAN2 (testCANclutch)	25.3	
<b>Engine Control</b>		
RPM	0	rpm
Intake air temp	-39.9	°C
Coolant temp	-39.9	°C
Throttle position	0.0	%
MAP	99.3	kPa
Battery voltage	12.10	V