Protokoll des CAN Test auf der Konsole:
Alle Eingaben auf der Konsole sind **blau** und **Fett** dargestellt!

```
schoene@schoene-TPad-X1:~/Dokumente/spmc-prj-ss17/hw-tests/can-test/atlys (master)
$ ll *.py
-rwxrwxr-x 1 schoene schoene 28990 Apr 26  2016 test_ACM0.py*
-rwxrwxr-x 1 schoene schoene 28994 Dez 21 13:57 test_XRUSB0.py*
schoene@schoene-TPad-X1:~/Dokumente/spmc-prj-ss17/hw-tests/can-test/atlys (master)
$ test_XRUSB0.py

Welcome to the SpartanMC <-> PCAN test script - Version Fr 11 Dez 2015 13:06:53 CET
This tools uses the SocketCAN API - see https://www.kernel.org/doc/Documentation/networking/can.txt for details.
It is important that the sudo command works on your machine as this script uses it.

Tested on:
> Ubuntu 12.04 LTS with Kernel 3.8.0-42-generic
> Ubuntu 12.04 LTS with Kernel 3.13.0-32-generic
> Raspbian with Kernel 3.12-1-rpi (Caution! More than 160 frames per test case cause problems with UART.)

Please close all programs that have access to the serial interface of the SpartanMC (e.g. minicom).
Press ENTER to continue...
+ The Atlys Board at /dev/ttyXRUSB0 is connected.

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 1

Before the PCAN interface is started, a bitrate has to be set.
[sudo] Passwort für schoene:
Interface disconnected.

0 - 1000kBit/s
1 -  500kBit/s
2 -  250kBit/s
3 -  125kBit/s
4 -  100kBit/s
5 -   50kBit/s
6 -   20kBit/s
7 -   10kBit/s
8 -    5kBit/s
9 -       Back

Please select the bitrate of the interface: 0
Bitrate set
Interface initialized.

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
```

```
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 9
m
 ungueltige Eingabe!

Einstellung der Datenrate und der Betriebsart
 Bus-Time 002AC CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :

t
Tx Buffer 18 Bit = 0x0
Einstellung der Datenrate und der Betriebsart
 Bus-Time 002AC CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :

n
Einstellung der Datenrate und der Betriebsart
 Bus-Time 002AC CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :
```

```
j


Bu 0B000 = 0F0F0 0AA55 0F0F0 0AA55 00000 28555 00000 03000
Bu 0B010 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00004 00000
Bu 0B020 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00000 00000
Bu 0B030 = 0F0F0 10000 00000 00000 3C3C3 12533 00000 00000
Bu 0B040 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 00001 00000
Bu 0B050 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 38533 00001 00000
Bu 0B060 = 00B00 00000 00000 00000 00000 01720 00000 00000
Bu 0B070 = 01122 03344 05566 07788 00000 00720 00000 00000
Bu 0B080 = 0F0F0 0AA55 0F0F0 0AA55 00000 21555 00000 03000
Bu 0B090 = 0CC33 0F0F0 0CC33 0F0F0 00000 0741F 00004 00000
Bu 0B0A0 = 0CC33 0F0F0 0CC33 0F0F0 00000 0641F 00004 00000
Bu 0B0B0 = 0F0F0 10000 00000 00000 3C3C3 10533 00000 00000
Bu 0B0C0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 17533 00001 00000
Bu 0B0D0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 37533 00001 00000
Bu 0B0E0 = 00B0C 00000 00000 00000 00000 02720 00000 00000
Bu 0B0F0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 16533 00001 00000
Bu 0B100 = 00B0C 00D00 00000 00000 00000 03720 00000 00000
Bu 0B110 = 00B0C 00D0E 00000 00000 00000 04720 00000 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

a


Naechster ISR-Puffer = 0 MAX ISR-Puffer = 150
tx-int: 00000000000 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 002AC
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00000  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
```

```
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 10
t
Tx Buffer 18 Bit = 0x0
Einstellung der Datenrate und der Betriebsart
 Bus-Time 002AC CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :

Please select the bitrate: 0
0
Einstellung der Datenrate und der Betriebsart
 Bus-Time 00151 CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :
```

```
Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 11
r

Das CAN Interface des SpartanMC fuehrt den Test mit
PCAN-longtime_tx vom Di 20 Dez 2016 08:25:38 CET aus.

Puffer anzeigen ? (j/n):n
Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 2
Wait for initialized SpartanMC
Please enter the test message in the form ID#DATA (Hex): 123#1234567890abcdef
Test-Message sent.

   508C 123#12.34.56.78.90.AB.CD.EF


Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
```

```
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 4
Wait for initialized SpartanMC
Please make sure that the SpartanMC is initialized via UART and the bitrate of PCAN and SpartanMC is the same!
Which predefined testcase should be replayed (for 1.1.case enter 1.1)? ascii
The script will now replay the test frames from file ascii.case
<open file 'Rx_2016-12-21_15-34-06.log', mode 'w' at 0x7f2ea8874930>
  can0  000   [3]  54 55 44                'TUD'
  can0  000   [7]  53 70 61 72 74 61 6E    'Spartan'
  can0  000   [5]  43 20 41 20 4E          'C A N'
  can0  000   [6]  72 6F 63 6B 73 21       'rocks!'

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 13
a


Naechster ISR-Puffer = 0 MAX ISR-Puffer = 150
tx-int: 000000000000 rx-int: 000000000004 ex-int: 000000000000
Tx Belegung 0x00000 Rx Belegung 0x20000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 000A8  CAN-tx-buff  = 00000  CAN-rx-buff  = 3C000

Bu 0B120 = 05455 04400 00000 00000 00000 03000 3FFF0 0178A
Bu 0B130 = 05370 06172 07461 06E00 00000 07000 3FFF0 02D08
Bu 0B140 = 04320 04120 04E00 00000 00000 05000 3FFF0 00A89
Bu 0B150 = 0726F 0636B 07321 00000 00000 06000 3FFF0 0474B
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
```

```
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 12
c

Die Rx Puffer werden geloescht

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000


Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
```

```
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 17
t
Tx Buffer 18 Bit = 0x3FFFF
Einstellung der Datenrate und der Betriebsart
 Bus-Time 00151 CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :


Das CAN Interface des SpartanMC fuehrt den Test mit
PCAN-longtime_tx vom Di 20 Dez 2016 08:25:38 CET aus.

Puffer anzeigen ? (j/n):

n
Anzahl der Tx-Telegramme eingeben:

144

  can0  555   [8]  remote request
  can0  41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                   '.'
  can0       720   [0]                       ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                       ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D             '...'
  can0       720   [4]  0B 0C 0D 0E          '....'
  can0       555   [8]  remote request
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                   '.'
  can0       720   [0]                       ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                       ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D             '...'
  can0       720   [4]  0B 0C 0D 0E          '....'
  can0       555   [8]  remote request
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                   '.'
  can0       720   [0]                       ''
```

```
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0       555   [8]  remote request
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0       555   [8]  remote request
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0       555   [8]  remote request
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0       555   [8]  remote request
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0       555   [8]  remote request
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
```

```
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                            ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                     '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D                  '...'
  can0       720   [4]  0B 0C 0D 0E               '....'

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

a

00000000001 Tx  1FFFF 0000
00000000002 Tx  0FFFF 0010
00000000003 Tx  07FFF 0020
00000000004 Tx  03FFF 0030
00000000005 Tx  01FFF 0040
00000000006 Tx  00FFF 0050
00000000007 Tx  007FF 0060
00000000008 Tx  003FF 0070
00000000009 Tx  001FF 0080
00000000010 Tx  000FF 0090
00000000011 Tx  0007F 00A0
00000000012 Tx  0003F 00B0
00000000013 Tx  0001F 00C0
00000000014 Tx  0000F 00D0
00000000015 Tx  00007 00E0
00000000016 Tx  00003 00F0
00000000017 Tx  00001 0100
00000000018 Tx  00000 0110
00000000019 Tx  1FFFF 0000
00000000020 Tx  0FFFF 0010
00000000021 Tx  07FFF 0020
00000000022 Tx  03FFF 0030
00000000023 Tx  01FFF 0040
00000000024 Tx  00FFF 0050
00000000025 Tx  007FF 0060
00000000026 Tx  003FF 0070
00000000027 Tx  001FF 0080
00000000028 Tx  000FF 0090
00000000029 Tx  0007F 00A0
00000000030 Tx  0003F 00B0
00000000031 Tx  0001F 00C0
00000000032 Tx  0000F 00D0
00000000033 Tx  00007 00E0
00000000034 Tx  00003 00F0
00000000035 Tx  00001 0100
00000000036 Tx  00000 0110
00000000037 Tx  1FFFF 0000
00000000038 Tx  0FFFF 0010
00000000039 Tx  07FFF 0020
00000000040 Tx  03FFF 0030
00000000041 Tx  01FFF 0040
00000000042 Tx  00FFF 0050
00000000043 Tx  007FF 0060
00000000044 Tx  003FF 0070
00000000045 Tx  001FF 0080
00000000046 Tx  000FF 0090
00000000047 Tx  0007F 00A0
00000000048 Tx  0003F 00B0
00000000049 Tx  0001F 00C0
00000000050 Tx  0000F 00D0
00000000051 Tx  00007 00E0
00000000052 Tx  00003 00F0
00000000053 Tx  00001 0100
00000000054 Tx  00000 0110
00000000055 Tx  1FFFF 0000
00000000056 Tx  0FFFF 0010
00000000057 Tx  07FFF 0020
00000000058 Tx  03FFF 0030
00000000059 Tx  01FFF 0040
00000000060 Tx  00FFF 0050
00000000061 Tx  007FF 0060
00000000062 Tx  003FF 0070
00000000063 Tx  001FF 0080
00000000064 Tx  000FF 0090
00000000065 Tx  0007F 00A0
00000000066 Tx  0003F 00B0
```

```
00000000067 Tx  0001F 00C0
00000000068 Tx  0000F 00D0
00000000069 Tx  00007 00E0
00000000070 Tx  00003 00F0
00000000071 Tx  00001 0100
00000000072 Tx  00000 0110
00000000073 Tx  1FFFF 0000
00000000074 Tx  0FFFF 0010
00000000075 Tx  07FFF 0020
00000000076 Tx  03FFF 0030
00000000077 Tx  01FFF 0040
00000000078 Tx  00FFF 0050
00000000079 Tx  007FF 0060
00000000080 Tx  003FF 0070
00000000081 Tx  001FF 0080
00000000082 Tx  000FF 0090
00000000083 Tx  0007F 00A0
00000000084 Tx  0003F 00B0
00000000085 Tx  0001F 00C0
00000000086 Tx  0000F 00D0
00000000087 Tx  00007 00E0
00000000088 Tx  00003 00F0
00000000089 Tx  00001 0100
00000000090 Tx  00000 0110
00000000091 Tx  1FFFF 0000
00000000092 Tx  0FFFF 0010
00000000093 Tx  07FFF 0020
00000000094 Tx  03FFF 0030
00000000095 Tx  01FFF 0040
00000000096 Tx  00FFF 0050
00000000097 Tx  007FF 0060
00000000098 Tx  003FF 0070
00000000099 Tx  001FF 0080
00000000100 Tx  000FF 0090
00000000101 Tx  0007F 00A0
00000000102 Tx  0003F 00B0
00000000103 Tx  0001F 00C0
00000000104 Tx  0000F 00D0
00000000105 Tx  00007 00E0
00000000106 Tx  00003 00F0
00000000107 Tx  00001 0100
00000000108 Tx  00000 0110
00000000109 Tx  1FFFF 0000
00000000110 Tx  0FFFF 0010
00000000111 Tx  07FFF 0020
00000000112 Tx  03FFF 0030
00000000113 Tx  01FFF 0040
00000000114 Tx  00FFF 0050
00000000115 Tx  007FF 0060
00000000116 Tx  003FF 0070
00000000117 Tx  001FF 0080
00000000118 Tx  000FF 0090
00000000119 Tx  0007F 00A0
00000000120 Tx  0003F 00B0
00000000121 Tx  0001F 00C0
00000000122 Tx  0000F 00D0
00000000123 Tx  00007 00E0
00000000124 Tx  00003 00F0
00000000125 Tx  00001 0100
00000000126 Tx  00000 0110
00000000127 Tx  1FFFF 0000
00000000128 Tx  0FFFF 0010
00000000129 Tx  07FFF 0020
00000000130 Tx  03FFF 0030
00000000131 Tx  01FFF 0040
00000000132 Tx  00FFF 0050
00000000133 Tx  007FF 0060
00000000134 Tx  003FF 0070
00000000135 Tx  001FF 0080
00000000136 Tx  000FF 0090
00000000137 Tx  0007F 00A0
00000000138 Tx  0003F 00B0
00000000139 Tx  0001F 00C0
00000000140 Tx  0000F 00D0
00000000141 Tx  00007 00E0
00000000142 Tx  00003 00F0
00000000143 Tx  00001 0100
00000000144 Tx  00000 0110

Naechster ISR-Puffer = 144 MAX ISR-Puffer = 150
tx-int: 00000000144 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x3FFFF Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00088
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
```

```
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 16
Please enter the test message in the form ID#DATA (Hex): 12345678#1234567890abcdef
  can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
a

00000000001 Tx  00000 0000

Naechster ISR-Puffer = 1 MAX ISR-Puffer = 150
tx-int: 000000000001 rx-int: 000000000000 ex-int: 000000000000
Tx Belegung 0x20000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000
```

```
Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 14
t



Bu 0B000 = 01234 05678 090AB 0CDEF 05678 1848D 00000 03000
Bu 0B010 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00004 00000
Bu 0B020 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00000 00000
Bu 0B030 = 0F0F0 10000 00000 00000 3C3C3 12533 00000 00000
Bu 0B040 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 00001 00000
Bu 0B050 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 38533 00001 00000
Bu 0B060 = 00B00 00000 00000 00000 00000 01720 00000 00000
Bu 0B070 = 01122 03344 05566 07788 00000 00720 00000 00000
Bu 0B080 = 0F0F0 0AA55 0F0F0 0AA55 00000 21555 00000 03000
Bu 0B090 = 0CC33 0F0F0 0CC33 0F0F0 00000 0741F 00004 00000
Bu 0B0A0 = 0CC33 0F0F0 0CC33 0F0F0 00000 0641F 00004 00000
Bu 0B0B0 = 0F0F0 10000 00000 00000 3C3C3 10533 00000 00000
Bu 0B0C0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 17533 00001 00000
Bu 0B0D0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 37533 00001 00000
Bu 0B0E0 = 00B0C 00000 00000 00000 00000 02720 00000 00000
Bu 0B0F0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 16533 00001 00000
Bu 0B100 = 00B0C 00D00 00000 00000 00000 03720 00000 00000
Bu 0B110 = 00B0C 00D0E 00000 00000 00000 04720 00000 00000
Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
```

```
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 17
t
Tx Buffer 18 Bit = 0x3FFFF
Einstellung der Datenrate und der Betriebsart
 Bus-Time 00151 CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :


Das CAN Interface des SpartanMC fuehrt den Test mit
PCAN-longtime_tx vom Di 20 Dez 2016 08:25:38 CET aus.

Puffer anzeigen ? (j/n):

n
Anzahl der Tx-Telegramme eingeben:

144

  can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                     '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                       '.'
  can0       720   [0]                           ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                           ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                    '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D                 '...'
  can0       720   [4]  0B 0C 0D 0E              '....'
  can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                     '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                       '.'
  can0       720   [0]                           ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                           ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                    '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D                 '...'
  can0       720   [4]  0B 0C 0D 0E              '....'
  can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                     '..'
```

```
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
can0  14CFC3C3   [8]  remote request
can0       720   [1]  0B                        '.'
can0       720   [0]                            ''
can0       555   [1]  remote request
can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
can0  14CFC3C3   [0]                            ''
can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
can0  14CFC3C3   [7]  remote request
can0       720   [2]  0B 0C                     '..'
can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
can0       720   [3]  0B 0C 0D                  '...'
can0       720   [4]  0B 0C 0D 0E               '....'
can0  12345678   [8]  12 34 56 78 90 AB CD EF   '.4Vx....'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0       41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
can0  14CFC3C3   [2]  F0 F0                     '..'
can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
```

```
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                        '.'
  can0       720   [0]                           ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                           ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                     '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D                  '...'
  can0       720   [4]  0B 0C 0D 0E               '....'

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

a

00000000001 Tx  1FFFF 0000
00000000002 Tx  0FFFF 0010
00000000003 Tx  07FFF 0020
00000000004 Tx  03FFF 0030
00000000005 Tx  01FFF 0040
00000000006 Tx  00FFF 0050
00000000007 Tx  007FF 0060
00000000008 Tx  003FF 0070
00000000009 Tx  001FF 0080
00000000010 Tx  000FF 0090
00000000011 Tx  0007F 00A0
00000000012 Tx  0003F 00B0
00000000013 Tx  0001F 00C0
00000000014 Tx  0000F 00D0
00000000015 Tx  00007 00E0
00000000016 Tx  00003 00F0
00000000017 Tx  00001 0100
00000000018 Tx  00000 0110
00000000019 Tx  1FFFF 0000
00000000020 Tx  0FFFF 0010
00000000021 Tx  07FFF 0020
00000000022 Tx  03FFF 0030
00000000023 Tx  01FFF 0040
00000000024 Tx  00FFF 0050
00000000025 Tx  007FF 0060
00000000026 Tx  003FF 0070
00000000027 Tx  001FF 0080
00000000028 Tx  000FF 0090
00000000029 Tx  0007F 00A0
00000000030 Tx  0003F 00B0
00000000031 Tx  0001F 00C0
00000000032 Tx  0000F 00D0
00000000033 Tx  00007 00E0
00000000034 Tx  00003 00F0
00000000035 Tx  00001 0100
00000000036 Tx  00000 0110
00000000037 Tx  1FFFF 0000
00000000038 Tx  0FFFF 0010
00000000039 Tx  07FFF 0020
00000000040 Tx  03FFF 0030
00000000041 Tx  01FFF 0040
00000000042 Tx  00FFF 0050
00000000043 Tx  007FF 0060
00000000044 Tx  003FF 0070
00000000045 Tx  001FF 0080
00000000046 Tx  000FF 0090
00000000047 Tx  0007F 00A0
00000000048 Tx  0003F 00B0
00000000049 Tx  0001F 00C0
00000000050 Tx  0000F 00D0
00000000051 Tx  00007 00E0
00000000052 Tx  00003 00F0
00000000053 Tx  00001 0100
00000000054 Tx  00000 0110
00000000055 Tx  1FFFF 0000
00000000056 Tx  0FFFF 0010
00000000057 Tx  07FFF 0020
00000000058 Tx  03FFF 0030
00000000059 Tx  01FFF 0040
00000000060 Tx  00FFF 0050
00000000061 Tx  007FF 0060
00000000062 Tx  003FF 0070
```

```
00000000063 Tx  001FF 0080
00000000064 Tx  000FF 0090
00000000065 Tx  0007F 00A0
00000000066 Tx  0003F 00B0
00000000067 Tx  0001F 00C0
00000000068 Tx  0000F 00D0
00000000069 Tx  00007 00E0
00000000070 Tx  00003 00F0
00000000071 Tx  00001 0100
00000000072 Tx  00000 0110
00000000073 Tx  1FFFF 0000
00000000074 Tx  0FFFF 0010
00000000075 Tx  07FFF 0020
00000000076 Tx  03FFF 0030
00000000077 Tx  01FFF 0040
00000000078 Tx  00FFF 0050
00000000079 Tx  007FF 0060
00000000080 Tx  003FF 0070
00000000081 Tx  001FF 0080
00000000082 Tx  000FF 0090
00000000083 Tx  0007F 00A0
00000000084 Tx  0003F 00B0
00000000085 Tx  0001F 00C0
00000000086 Tx  0000F 00D0
00000000087 Tx  00007 00E0
00000000088 Tx  00003 00F0
00000000089 Tx  00001 0100
00000000090 Tx  00000 0110
00000000091 Tx  1FFFF 0000
00000000092 Tx  0FFFF 0010
00000000093 Tx  07FFF 0020
00000000094 Tx  03FFF 0030
00000000095 Tx  01FFF 0040
00000000096 Tx  00FFF 0050
00000000097 Tx  007FF 0060
00000000098 Tx  003FF 0070
00000000099 Tx  001FF 0080
00000000100 Tx  000FF 0090
00000000101 Tx  0007F 00A0
00000000102 Tx  0003F 00B0
00000000103 Tx  0001F 00C0
00000000104 Tx  0000F 00D0
00000000105 Tx  00007 00E0
00000000106 Tx  00003 00F0
00000000107 Tx  00001 0100
00000000108 Tx  00000 0110
00000000109 Tx  1FFFF 0000
00000000110 Tx  0FFFF 0010
00000000111 Tx  07FFF 0020
00000000112 Tx  03FFF 0030
00000000113 Tx  01FFF 0040
00000000114 Tx  00FFF 0050
00000000115 Tx  007FF 0060
00000000116 Tx  003FF 0070
00000000117 Tx  001FF 0080
00000000118 Tx  000FF 0090
00000000119 Tx  0007F 00A0
00000000120 Tx  0003F 00B0
00000000121 Tx  0001F 00C0
00000000122 Tx  0000F 00D0
00000000123 Tx  00007 00E0
00000000124 Tx  00003 00F0
00000000125 Tx  00001 0100
00000000126 Tx  00000 0110
00000000127 Tx  1FFFF 0000
00000000128 Tx  0FFFF 0010
00000000129 Tx  07FFF 0020
00000000130 Tx  03FFF 0030
00000000131 Tx  01FFF 0040
00000000132 Tx  00FFF 0050
00000000133 Tx  007FF 0060
00000000134 Tx  003FF 0070
00000000135 Tx  001FF 0080
00000000136 Tx  000FF 0090
00000000137 Tx  0007F 00A0
00000000138 Tx  0003F 00B0
00000000139 Tx  0001F 00C0
00000000140 Tx  0000F 00D0
00000000141 Tx  00007 00E0
00000000142 Tx  00003 00F0
00000000143 Tx  00001 0100
00000000144 Tx  00000 0110

Naechster ISR-Puffer = 144 MAX ISR-Puffer = 150
tx-int: 00000000144 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x3FFFF Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00088
```

```
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 18
Please make sure that the SpartanMC is initialized via UART and the bitrate of PCAN and SpartanMC is the same!
Which predefined testcase should be replayed (for 1.1.case enter 1.1)? 2
The script will now replay the test frames from file 2.case
<open file 'Tx_2016-12-21_15-45-32.log', mode 'w' at 0x7f2ea8874ae0>
1       (1.0) can0 555#R8
2       (1.1) can0 41F#CC.33.F0.F0.CC.33.F0
3       (1.2) can0 41F#CC.33.F0.F0.CC.33.F0.F0
4       (1.3) can0 14CFC3C3#F0.F0
5       (1.4) can0 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
6       (1.5) can0 14CFC3C3#R8
7       (1.6) can0 720#0B
8       (1.7) can0 720#
9       (1.8) can0 555#R1
10      (1.9) can0 41F#CC.33.F0.F0.CC.33.F0
11      (2.0) can0 41F#CC.33.F0.F0.CC.33
12      (2.1) can0 14CFC3C3#
13      (2.2) can0 14CFC3C3#F0.F0.F0.F0.CC.33.AA
14      (2.3) can0 14CFC3C3#R7
15      (2.4) can0 720#0B.0C
16      (2.5) can0 14CFC3C3#F0.F0.F0.F0.CC.33
17      (2.6) can0 720#0B.0C.0D
18      (2.7) can0 720#0B.0C.0D.0E

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
```

```
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 13
a


Naechster ISR-Puffer = 0 MAX ISR-Puffer = 150
tx-int: 00000000018 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x20000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit
```

```
Please select: 12
c

Die Rx Puffer werden geloescht

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000


Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 18
Please make sure that the SpartanMC is initialized via UART and the bitrate of PCAN and SpartanMC is the same!
Which predefined testcase should be replayed (for 1.1.case enter 1.1)? ascii
The script will now replay the test frames from file ascii.case
<open file 'Tx_2016-12-21_15-48-12.log', mode 'w' at 0x7f2ea8874d20>
1       (1.0) can0 000#545544
2       (2.0) can0 000#5370617274616e
3       (3.0) can0 000#432041204e
4       (4.0) can0 000#726f636b7321

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
```

```
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 13
a


Naechster ISR-Puffer = 0 MAX ISR-Puffer = 150
tx-int: 00000000004 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x20000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 12
c

Die Rx Puffer werden geloescht

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
```

```
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000


Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 19
Set SpartanMC ACF-enable to 0x0000f
Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 4
Wait for initialized SpartanMC
Please make sure that the SpartanMC is initialized via UART and the bitrate of PCAN and SpartanMC is the same!
Which predefined testcase should be replayed (for 1.1.case enter 1.1)? 2
The script will now replay the test frames from file 2.case
<open file 'Rx_2016-12-21_15-52-16.log', mode 'w' at 0x7f2ea8874930>
  can0  555   [8]  remote request
  can0  41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0  41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
```

```
  can0  14CFC3C3   [2]  F0 F0                     '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                       '.'
  can0       720   [0]                          ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                          ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                    '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D                 '...'
  can0       720   [4]  0B 0C 0D 0E              '....'

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 13
a


Naechster ISR-Puffer = 0 MAX ISR-Puffer = 150
tx-int: 00000000000 rx-int: 00000000018 ex-int: 00000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 0000F
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00118  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 28555 00002 0608E
Bu 0B130 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 00001 05A40
Bu 0B140 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00001 06A52
Bu 0B150 = 0F0F0 00000 00000 00000 3C3C3 12533 00008 007AA
Bu 0B160 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 00008 029D2
Bu 0B170 = 00000 00000 00000 00000 3C3C3 38533 00008 0400D
Bu 0B180 = 00B00 00000 00000 00000 00000 01720 00004 07444
Bu 0B190 = 00000 00000 00000 00000 00000 00720 00004 05BC8
Bu 0B1A0 = 00000 00000 00000 00000 00000 21555 00002 05110
Bu 0B1B0 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 00001 05A40
Bu 0B1C0 = 0CC33 0F0F0 0CC33 00000 00000 0641F 00001 03F4D
Bu 0B1D0 = 00000 00000 00000 00000 3C3C3 10533 00008 047CF
Bu 0B1E0 = 0F0F0 0F0F0 0CC33 0AA00 3C3C3 17533 00008 06A4F
Bu 0B1F0 = 00000 00000 00000 00000 3C3C3 37533 00008 067F7
Bu 0B200 = 00B0C 00000 00000 00000 00000 02720 00004 04392
Bu 0B210 = 0F0F0 0F0F0 0CC33 00000 3C3C3 16533 00008 0438F
Bu 0B220 = 00B0C 00D00 00000 00000 00000 03720 00004 01447
Bu 0B230 = 00B0C 00D0E 00000 00000 00000 04720 00004 07379

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********
```

```
00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 12
c

Die Rx Puffer werden geloescht

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000


Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 20
Set SpartanMC ACF-enable to 0x00000
Main Menu
*********
```

```
00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 4
Wait for initialized SpartanMC
Please make sure that the SpartanMC is initialized via UART and the bitrate of PCAN and SpartanMC is the same!
Which predefined testcase should be replayed (for 1.1.case enter 1.1)? 2
The script will now replay the test frames from file 2.case
<open file 'Rx_2016-12-21_15-55-13.log', mode 'w' at 0x7f2ea8874930>
  can0  555   [8]  remote request
  can0  41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0  41F   [8]  CC 33 F0 F0 CC 33 F0 F0   '.3...3..'
  can0  14CFC3C3   [2]  F0 F0                '..'
  can0  14CFC3C3   [8]  F0 F0 F0 F0 CC 33 AA 55   '.....3.U'
  can0  14CFC3C3   [8]  remote request
  can0       720   [1]  0B                   '.'
  can0       720   [0]                       ''
  can0       555   [1]  remote request
  can0       41F   [7]  CC 33 F0 F0 CC 33 F0      '.3...3.'
  can0       41F   [6]  CC 33 F0 F0 CC 33         '.3...3'
  can0  14CFC3C3   [0]                       ''
  can0  14CFC3C3   [7]  F0 F0 F0 F0 CC 33 AA      '.....3.'
  can0  14CFC3C3   [7]  remote request
  can0       720   [2]  0B 0C                '..'
  can0  14CFC3C3   [6]  F0 F0 F0 F0 CC 33         '.....3'
  can0       720   [3]  0B 0C 0D             '...'
  can0       720   [4]  0B 0C 0D 0E          '....'

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 13
a


Naechster ISR-Puffer = 0 MAX ISR-Puffer = 150
tx-int: 00000000000 rx-int: 00000000018 ex-int: 00000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00118  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 28555 3FFF2 0608E
Bu 0B130 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 3FFF1 05A40
```

```
Bu 0B140 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 3FFF1 06A52
Bu 0B150 = 0F0F0 00000 00000 00000 3C3C3 12533 3FFF8 007AA
Bu 0B160 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 3FFF8 029D2
Bu 0B170 = 00000 00000 00000 00000 3C3C3 38533 3FFF8 0400D
Bu 0B180 = 00B00 00000 00000 00000 00000 01720 3FFF4 07444
Bu 0B190 = 00000 00000 00000 00000 00000 00720 3FFF4 05BC8
Bu 0B1A0 = 00000 00000 00000 00000 00000 21555 3FFF2 05110
Bu 0B1B0 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 3FFF1 05A40
Bu 0B1C0 = 0CC33 0F0F0 0CC33 00000 00000 0641F 3FFF1 03F4D
Bu 0B1D0 = 00000 00000 00000 00000 3C3C3 10533 3FFF8 047CF
Bu 0B1E0 = 0F0F0 0F0F0 0CC33 0AA00 3C3C3 17533 3FFF8 06A4F
Bu 0B1F0 = 00000 00000 00000 00000 3C3C3 37533 3FFF8 067F7
Bu 0B200 = 00B0C 00000 00000 00000 00000 02720 3FFF4 04392
Bu 0B210 = 0F0F0 0F0F0 0CC33 00000 3C3C3 16533 3FFF8 0438F
Bu 0B220 = 00B0C 00D00 00000 00000 00000 03720 3FFF4 01447
Bu 0B230 = 00B0C 00D0E 00000 00000 00000 04720 3FFF4 07379

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 12
c

Die Rx Puffer werden geloescht

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000


Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
```

```
d - DEBUG off
Eingabe:

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 21
schoene@schoene-TPad-X1:~/Dokumente/spmc-prj-ss17/hw-tests/can-test/atlys (master)
$
```

Protokoll eines 2. SpartanMC mit CAN Interface im Listenmodus auf dem Minicom:

Auch hier werden alle Eingaben auf dem Minicom **blau** und **Fett** dargestellt!

```
Stack = 0AFF0

Das CAN Interface des SpartanMC fuehrt den Test mit
PCAN-longtime_tx vom Di 20 Dez 2016 08:25:38 CET aus.


Einstellung der Datenrate und der Betriebsart
 Bus-Time 002AC CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 : 0
Einstellung der Datenrate und der Betriebsart
 Bus-Time 00151 CAN-Mode 00070

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
```

```
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 : l
Einstellung der Datenrate und der Betriebsart
 Bus-Time 00151 CAN-Mode 00270

 0 - 1000kBit/s
 1 -  500kBit/s
 2 -  250kBit/s
 3 -  125kBit/s
 4 -  100kBit/s
 5 -   50kBit/s
 6 -   20kBit/s
 7 -   10kBit/s
 8 -    5kBit/s
 9 - Bus-Timing Hex eingeben

 t - Tx Buffer Hex aktivieren
 d - Daten in Tx Buffer schreiben

 a - ACF-Filter eistellen
 f - ACF-Filter freigeben

 s - CAN-Mode Selbstest
 n - CAN-Mode Normal
 l - CAN-Mode Listen only
ESC- Ende der CAN-Einstellungen
 :
Das CAN Interface des SpartanMC fuehrt den Test mit
PCAN-longtime_tx vom Di 20 Dez 2016 08:25:38 CET aus.

Puffer anzeigen ? (j/n):j


Bu 0B000 = 0F0F0 0AA55 0F0F0 0AA55 00000 28555 00000 03000
Bu 0B010 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00004 00000
Bu 0B020 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 00000 00000
Bu 0B030 = 0F0F0 10000 00000 00000 3C3C3 12533 00000 00000
Bu 0B040 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 00001 00000
Bu 0B050 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 38533 00001 00000
Bu 0B060 = 00B00 00000 00000 00000 00000 01720 00000 00000
Bu 0B070 = 01122 03344 05566 07788 00000 00720 00000 00000
Bu 0B080 = 0F0F0 0AA55 0F0F0 0AA55 00000 21555 00000 03000
Bu 0B090 = 0CC33 0F0F0 0CC33 0F0F0 00000 0741F 00004 00000
Bu 0B0A0 = 0CC33 0F0F0 0CC33 0F0F0 00000 0641F 00004 00000
Bu 0B0B0 = 0F0F0 10000 00000 00000 3C3C3 10533 00000 00000
Bu 0B0C0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 17533 00001 00000
Bu 0B0D0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 37533 00001 00000
Bu 0B0E0 = 00B0C 00000 00000 00000 00000 02720 00000 00000
Bu 0B0F0 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 16533 00001 00000
Bu 0B100 = 00B0C 00D00 00000 00000 00000 03720 00000 00000
Bu 0B110 = 00B0C 00D0E 00000 00000 00000 04720 00000 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:    508C 123#12.34.56.78.90.AB.CD.EF
   178A 000#54.55.44
   2D08 000#53.70.61.72.74.61.6E
   0A89 000#43.20.41.20.4E
```

```
474B 000#72.6F.63.6B.73.21
608E 555#R8
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
07AA 14CFC3C3#F0.F0
29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
400D 14CFC3C3#R8
7444 720#0B
5BC8 720#
5110 555#R1
5A40 41F#CC.33.F0.F0.CC.33.F0
3F4D 41F#CC.33.F0.F0.CC.33
47CF 14CFC3C3#
6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
67F7 14CFC3C3#R7
4392 720#0B.0C
438F 14CFC3C3#F0.F0.F0.F0.CC.33
1447 720#0B.0C.0D
7379 720#0B.0C.0D.0E
608E 555#R8
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
438F 14CFC3C3#F0.F0.F0.F0.CC.33
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
7444 720#0B
7379 720#0B.0C.0D.0E
1679 12345678#12.34.56.78.90.AB.CD.EF
1679 12345678#12.34.56.78.90.AB.CD.EF
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
07AA 14CFC3C3#F0.F0
29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
400D 14CFC3C3#R8
7444 720#0B
5BC8 720#
5110 555#R1
5A40 41F#CC.33.F0.F0.CC.33.F0
3F4D 41F#CC.33.F0.F0.CC.33
47CF 14CFC3C3#
6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
67F7 14CFC3C3#R7
4392 720#0B.0C
438F 14CFC3C3#F0.F0.F0.F0.CC.33
1447 720#0B.0C.0D
7379 720#0B.0C.0D.0E
1679 12345678#12.34.56.78.90.AB.CD.EF
5110 555#R1
7444 720#0B
29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
4392 720#0B.0C
1447 720#0B.0C.0D
7379 720#0B.0C.0D.0E
608E 555#R8
5A40 41F#CC.33.F0.F0.CC.33.F0
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
07AA 14CFC3C3#F0.F0
29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
400D 14CFC3C3#R8
7444 720#0B
5BC8 720#
5110 555#R1
5A40 41F#CC.33.F0.F0.CC.33.F0
3F4D 41F#CC.33.F0.F0.CC.33
47CF 14CFC3C3#
6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
67F7 14CFC3C3#R7
4392 720#0B.0C
438F 14CFC3C3#F0.F0.F0.F0.CC.33
1447 720#0B.0C.0D
7379 720#0B.0C.0D.0E
178A 000#54.55.44
2D08 000#53.70.61.72.74.61.6E
0A89 000#43.20.41.20.4E
474B 000#72.6F.63.6B.73.21
608E 555#R8
5A40 41F#CC.33.F0.F0.CC.33.F0
6A52 41F#CC.33.F0.F0.CC.33.F0.F0
07AA 14CFC3C3#F0.F0
29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
400D 14CFC3C3#R8
7444 720#0B
5BC8 720#
5110 555#R1
5A40 41F#CC.33.F0.F0.CC.33.F0
3F4D 41F#CC.33.F0.F0.CC.33
47CF 14CFC3C3#
6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
67F7 14CFC3C3#R7
4392 720#0B.0C
438F 14CFC3C3#F0.F0.F0.F0.CC.33
1447 720#0B.0C.0D
```

```
   7379 720#0B.0C.0D.0E
   608E 555#R8
   5A40 41F#CC.33.F0.F0.CC.33.F0
   6A52 41F#CC.33.F0.F0.CC.33.F0.F0
   07AA 14CFC3C3#F0.F0
   29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
   400D 14CFC3C3#R8
   7444 720#0B
   5BC8 720#
   5110 555#R1
   5A40 41F#CC.33.F0.F0.CC.33.F0
   3F4D 41F#CC.33.F0.F0.CC.33
   47CF 14CFC3C3#
   6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
   67F7 14CFC3C3#R7
   4392 720#0B.0C
   438F 14CFC3C3#F0.F0.F0.F0.CC.33
   1447 720#0B.0C.0D
   7379 720#0B.0C.0D.0E

CTRL-A Z for help | 115200 8N2 | NOR | Minicom 2.7 | VT102 | Online 1:33 | ttyXRUSB1
```

3. Hier noch einige Bilder vom Minicom und der Testkonsole während der Tests:

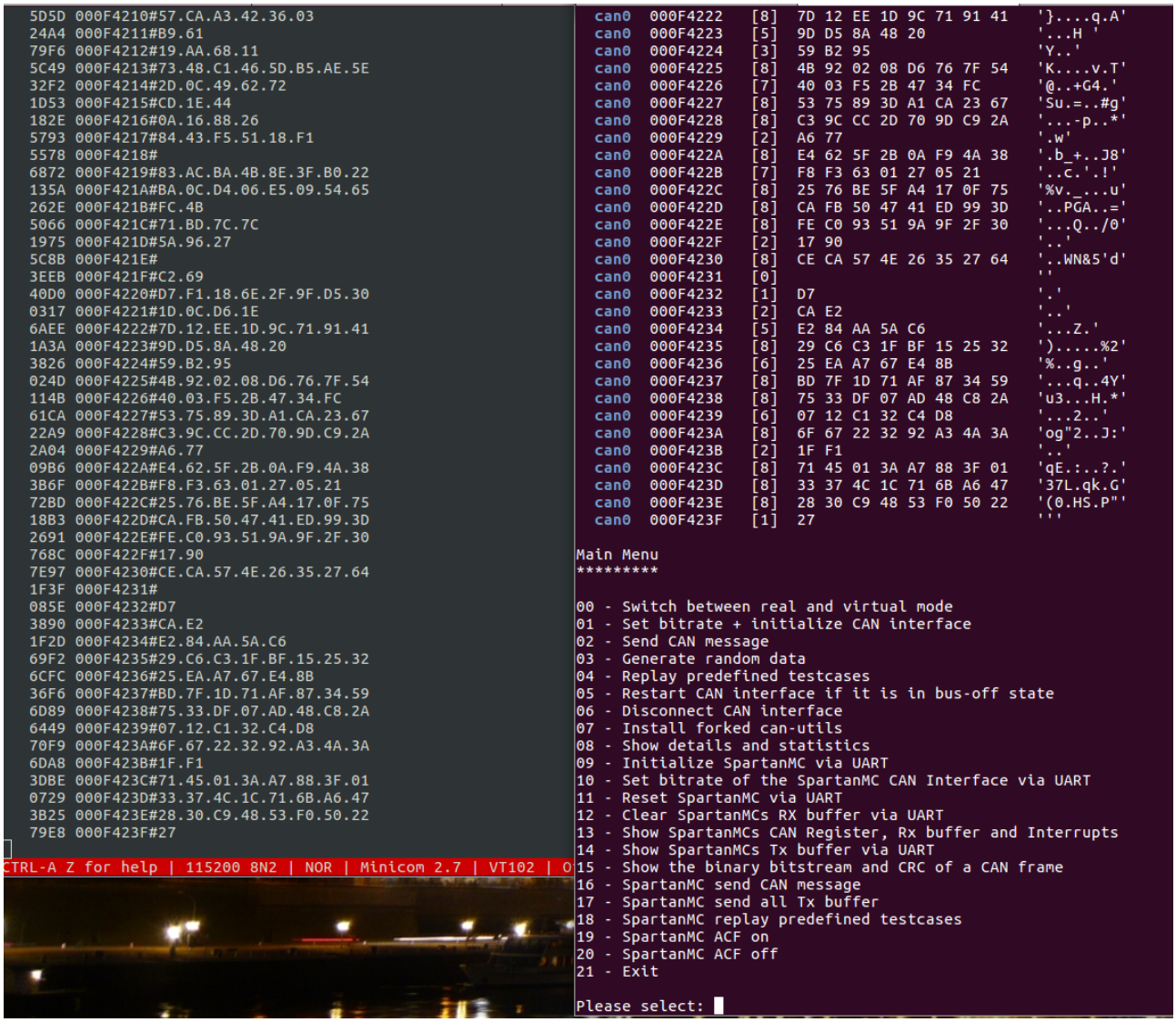


Bild 1: Minicom und Konsole nach dem Start des Test 04 auf der Konsole mit der Datei 4.2.case mit 1000000 Datensätzen mit zufälliger Länge und zufälligen Daten.

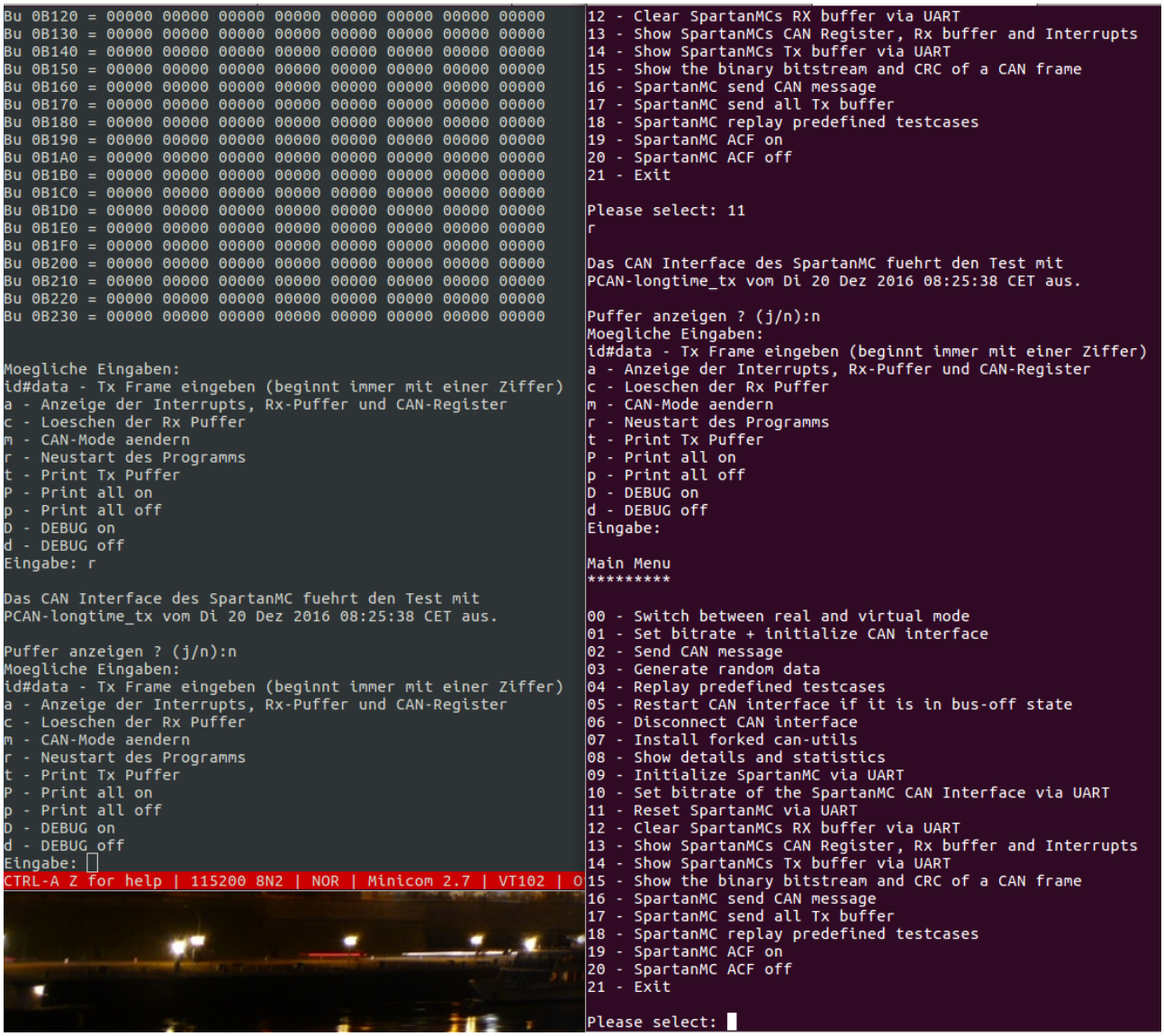


Bild 2: RESET für den SpartanMC auf der Konsole mit Kommando 11 auslösen und auch den SpartanMC im Listen Mode auf dem Minicom mit Kommando r rücksetzen.

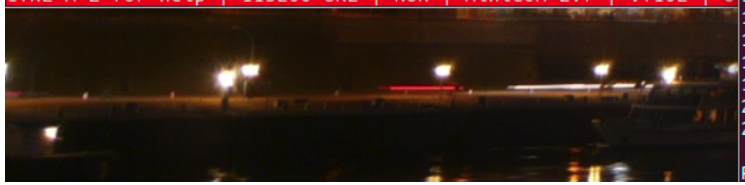
Auf der nächsten Seite ist im Bild 3 der Start von Test 04 mit der Datei ascii.case und die dabei erfolgten Ausgaben auf der Konsole und dem Minicom zu sehen. Auf dem Minicom wurden dabei nur die letzten 4 Zeilen mit dem Protokoll der 4 Nachrichten aus ascii.case angezeigt.

```
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000


Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe: r

Das CAN Interface des SpartanMC fuehrt den Test mit
PCAN-longtime_tx vom Di 20 Dez 2016 08:25:38 CET aus.

Puffer anzeigen ? (j/n):n
Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:    178A 000#54.55.44
   2D08 000#53.70.61.72.74.61.6E
   0A89 000#43.20.41.20.4E
   474B 000#72.6F.63.6B.73.21
CTRL-A Z for help | 115200 8N2 | NOR | Minicom 2.7 | VT102 | O
```



```
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select: 4
Wait for initialized SpartanMC
Please make sure that the SpartanMC is initialized via UART and the bitrate of PCAN and SpartanMC is the same!
Which predefined testcase should be replayed (for 1.1.case enter 1.1)? ascii
The script will now replay the test frames from file ascii.case
<open file 'Rx_2016-12-21_11-22-01.log', mode 'w' at 0x7effb53a0930>
  can0  000   [3]  54 55 44                'TUD'
  can0  000   [7]  53 70 61 72 74 61 6E    'Spartan'
  can0  000   [5]  43 20 41 20 4E          'C A N'
  can0  000   [6]  72 6F 63 6B 73 21       'rocks!'

Main Menu
*********

00 - Switch between real and virtual mode
01 - Set bitrate + initialize CAN interface
02 - Send CAN message
03 - Generate random data
04 - Replay predefined testcases
05 - Restart CAN interface if it is in bus-off state
06 - Disconnect CAN interface
07 - Install forked can-utils
08 - Show details and statistics
09 - Initialize SpartanMC via UART
10 - Set bitrate of the SpartanMC CAN Interface via UART
11 - Reset SpartanMC via UART
12 - Clear SpartanMCs RX buffer via UART
13 - Show SpartanMCs CAN Register, Rx buffer and Interrupts
14 - Show SpartanMCs Tx buffer via UART
15 - Show the binary bitstream and CRC of a CAN frame
16 - SpartanMC send CAN message
17 - SpartanMC send all Tx buffer
18 - SpartanMC replay predefined testcases
19 - SpartanMC ACF on
20 - SpartanMC ACF off
21 - Exit

Please select:
```

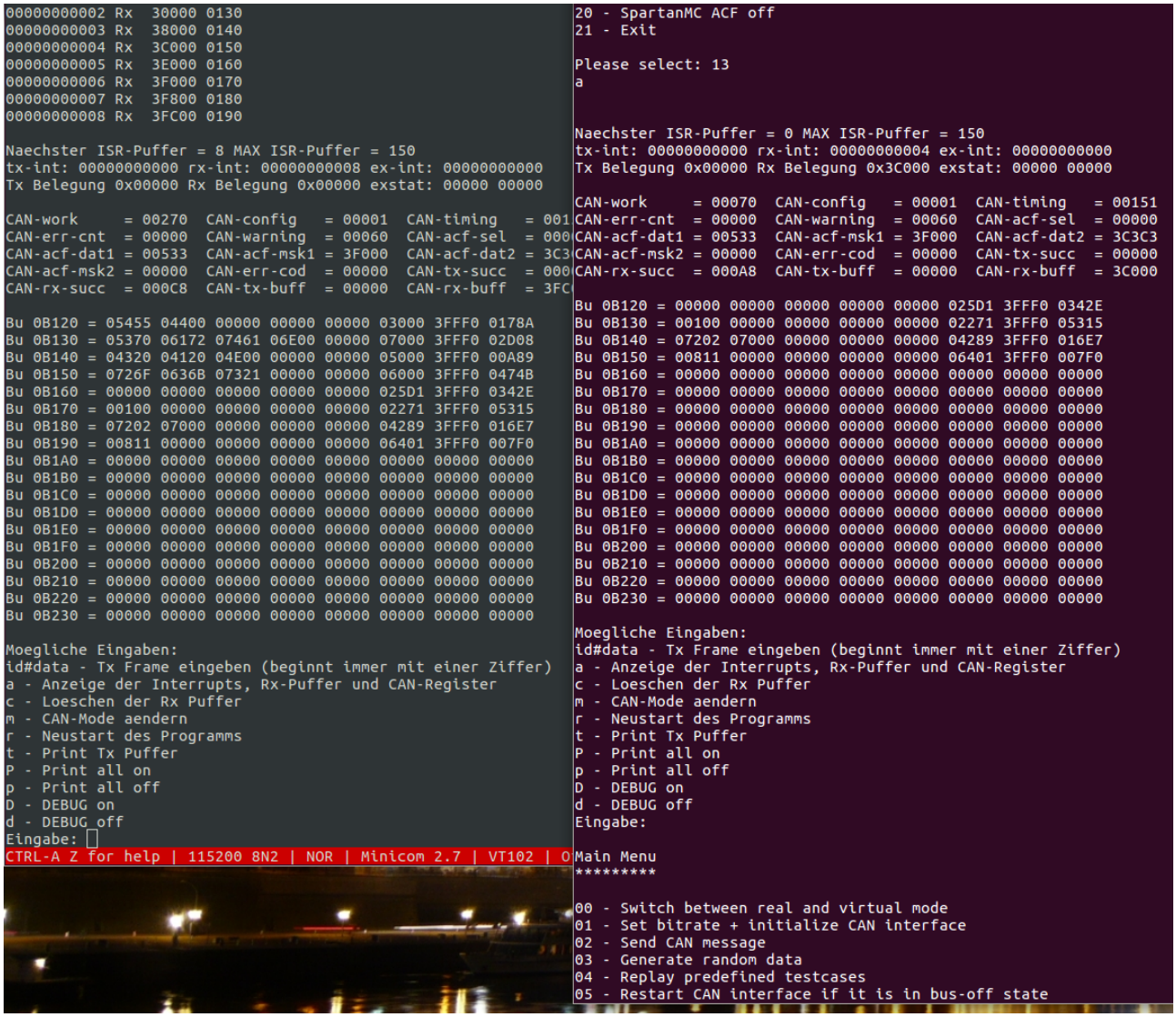


Bild 4: Konsole nach der Ausführung von Kommando 04 mit der Datei test.case und danach das Kommando 13 zur Anzeige der Ergebnisse des Tests. Es wurden 4 Rx Inerrupts ausgeführt und die empfangenen Datensätze aus test.case befinden sich in den ersten 4 Empfangspuffern des SpartanMC.
Auf dem Minicom wurde mit der Eingabe von a auch die Protokollierung der letzten empfangen Daten aufgerufen. Dies erfolgte aber erst nach dem Start von Kommando 4 mit ascii.case und danach Kommando 4 mit test.case auf der Konsole. Deshalb wurden auch 8 Nachrichten empfangen, welche sich in den ersten 8 Puffern dieses SpartanMC befinden. Zuerst die 4 Nachrichten aus ascii.case und danach die 4 Nachrichten aus test.case. In der ganz rechten Spalte der jeweils angezeigten Tabelle befindet sich immer der CRC der Nachricht in dieser Zeile.

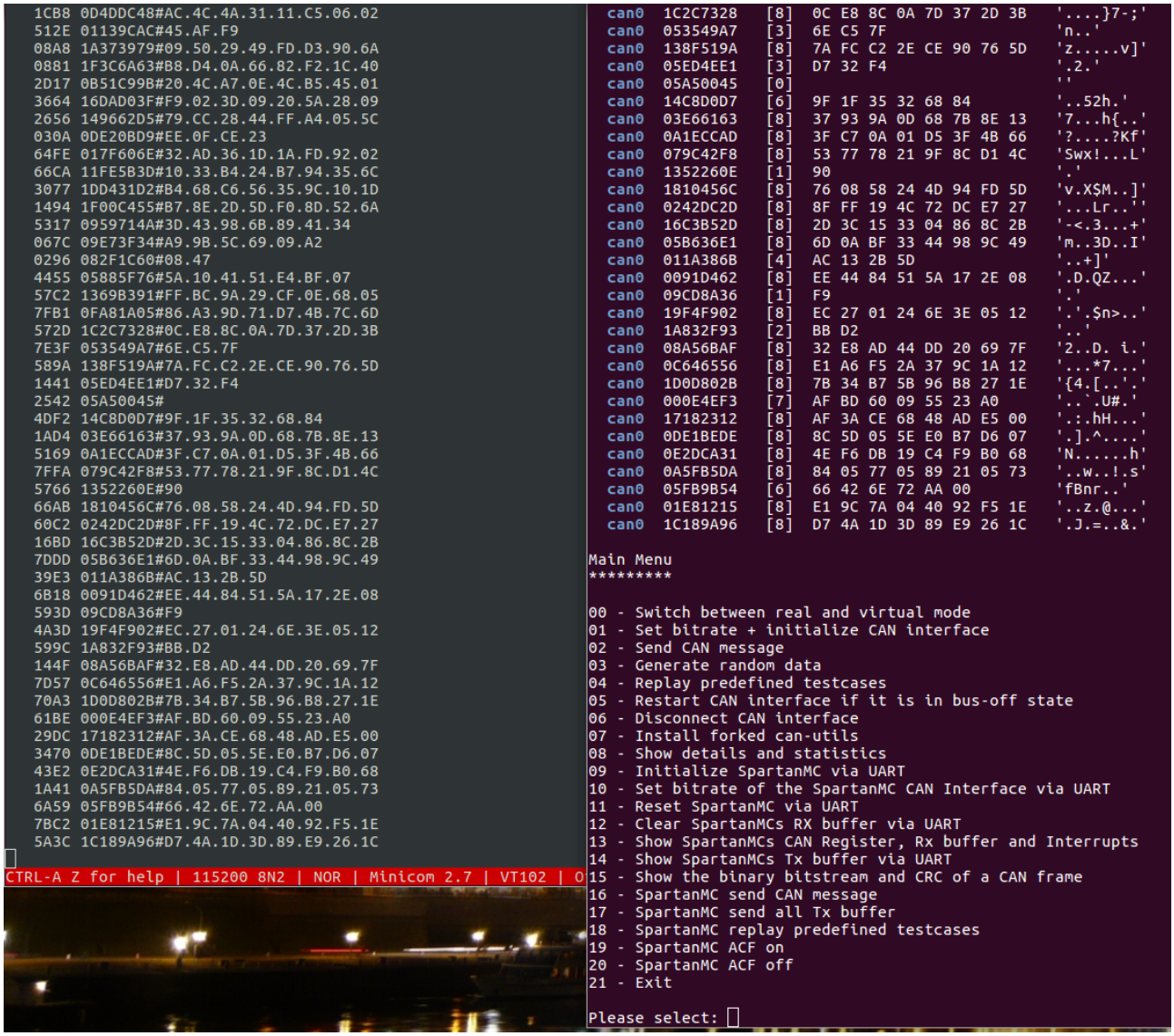


Bild 5: Konsole und Minicom nach dem Beenden von Kommando 04 auf der Konsole mit der Datei 5.case.

4. Aufbau der bei den Tests erstellten *.log Dateien

Es wird als erstes eine *.log Datei dargestellt, die beim Empfang der Nachrichten aus ascii.case auf dem SpartanMC entstanden ist. In diese Darstellung sind Kommentare zur Erläuterung der Darstellungen eingebaut. Danach kommt dann noch einmal die gleiche Datei ohne Kommentare.

Datei: Rx_2016-12-21_11-22-01.log mit Kommentaren.

Die Datei beginnt mit der Startzeit des Tests und der verwendeten *.case Datei.

```
PCAN -> SpartanMC test in replay mode - run on 2016-12-21 11:22:01    ascii.case
```

Es folgen die Sende und Empfangsprotokolle mit der Fehlerauswertung der 4 Nachrichten und am Ende der Summe aller Fehler.

```
cansend: 178A 000#54.55.44
spartan: 178A 000#54.55.44
FRAME 000001 OK

cansend: 2D08 000#53.70.61.72.74.61.6E
spartan: 2D08 000#53.70.61.72.74.61.6E
FRAME 000002 OK

cansend: 0A89 000#43.20.41.20.4E
spartan: 0A89 000#43.20.41.20.4E
FRAME 000003 OK

cansend: 474B 000#72.6F.63.6B.73.21
spartan: 474B 000#72.6F.63.6B.73.21
FRAME 000004 OK

Error Cnt = 000000
```

Der Test löst zum Schluss durch Senden von a noch die Protokollierung vom SpartanMC auf. Es können die letzten 150 Interrupts protokolliert werden und dazu immer der Inhalt des CAN-rx-buff Registers und die Adresse des Puffers. Es folgen dann die Anzahl aller ausgeführten Interrupts, Programm Status Informationen, die CAN-Register und die 18 Rx Puffer des SpartanMC. Zum Schluss wird noch einmal die Zeit beim Abschluss des Tests eingeblendet. Mit den farbigen Darstellungen soll der Zusammenhang mit den Daten der Nachrichten und die Zuordnung im Rx Puffer hergestellt werden.

```
a

00000000001 Rx  20000 0120
00000000002 Rx  30000 0130
00000000003 Rx  38000 0140
00000000004 Rx  3C000 0150

Naechster ISR-Puffer = 4 MAX ISR-Puffer = 150
tx-int: 00000000000 rx-int: 00000000004 ex-int: 00000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 000A8  CAN-tx-buff  = 00000  CAN-rx-buff  = 3C000

Bu 0B120 = 05455 04400 00000 00000 00000 03000 3FFF0 0178A
Bu 0B130 = 05370 06172 07461 06E00 00000 07000 3FFF0 02D08
Bu 0B140 = 04320 04120 04E00 00000 00000 05000 3FFF0 00A89
Bu 0B150 = 0726F 0636B 07321 00000 00000 06000 3FFF0 0474B
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:
PCAN -> SpartanMC test in replay mode - end on 2016-12-21 11:22:04    ascii.case
Error Cnt = 000000
```

In der 2. Spalte von rechts in der obigen Tabelle werden die Ausgangssignale aller 18 Filter angezeigt. Alle nicht initialisierten Filter erzeugen dabei immer eine 1. In den unteren 4 Bit werden die Ergebnisse der 4 initialisierten Filter angezeigt. Da der ID 000 in keinem der Filter steht, sind diese 4 Bit bei allen Nachrichten 0.

Datei: Rx_2016-12-21_11-22-01.log ohne Kommentare

```
PCAN -> SpartanMC test in replay mode - run on 2016-12-21 11:22:01    ascii.case

cansend: 178A 000#54.55.44
spartan: 178A 000#54.55.44
FRAME 000001 OK

cansend: 2D08 000#53.70.61.72.74.61.6E
spartan: 2D08 000#53.70.61.72.74.61.6E
FRAME 000002 OK

cansend: 0A89 000#43.20.41.20.4E
spartan: 0A89 000#43.20.41.20.4E
FRAME 000003 OK

cansend: 474B 000#72.6F.63.6B.73.21
spartan: 474B 000#72.6F.63.6B.73.21
FRAME 000004 OK

Error Cnt = 000000

a

00000000001 Rx  20000 0120
00000000002 Rx  30000 0130
00000000003 Rx  38000 0140
00000000004 Rx  3C000 0150

Naechster ISR-Puffer = 4 MAX ISR-Puffer = 150
tx-int: 00000000000 rx-int: 00000000004 ex-int: 00000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 000A8  CAN-tx-buff  = 00000  CAN-rx-buff  = 3C000

Bu 0B120 = 05455 04400 00000 00000 00000 03000 3FFF0 0178A
Bu 0B130 = 05370 06172 07461 06E00 00000 07000 3FFF0 02D08
Bu 0B140 = 04320 04120 04E00 00000 00000 05000 3FFF0 00A89
Bu 0B150 = 0726F 0636B 07321 00000 00000 06000 3FFF0 0474B
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:
PCAN -> SpartanMC test in replay mode - end on 2016-12-21 11:22:04    ascii.case
Error Cnt = 000000
```

Datei: Rx_2016-12-20_15-08-04.log mit Kommentareen. Da hier die ID der Nachrichten mit den Werten in den Filtern übereinstimmen, erscheint auch bei dem jeweiligen Filter eine 1 in den unteren 4 Bit der vorletzten Spalte. Mit den farbigen Darstellungen soll der Zusammenhang mit den Daten der Nachrichten und die Zuordnung im Rx Puffer hergestellt werden. Mit der Farbe Grün werden dabei zwei Informationen im Rx Puffer gekennzeichnet. Das niedrigste Bit legt fest, dass diese Nachricht einen erweiterten ID von 29 Bit hat. Deshalb ist dieses Bit auch in den Beiden Nachrichten in den Puffern darüber gesetzt. Das höhere Bit kennzeichnet die Nachricht als „Remote Frame“. Danach steht mit blau immer die Länge der Nachricht von 0 bis 8. Nach der Länge folgen die oberen 11 Bit vom ID in violett und die unteren 18 Bit stehen in der Spalte links daneben.

```
 5    3    3      3   C    3    C    3
101 0011 0011    11 1100 0011 1100 0011  und nach dem Zusammenfügen der 11 und 18 Bit
1 0100 1100    1111 1100 0011 1100 0011
1   4    C       F    C    3    C    3

PCAN -> SpartanMC test in replay mode - run on 2016-12-20 15:08:04    2.case

cansend: 608E 555#R8
spartan: 608E 555#R8
FRAME 000001 OK

cansend: 5A40 41F#CC.33.F0.F0.CC.33.F0
spartan: 5A40 41F#CC.33.F0.F0.CC.33.F0
FRAME 000002 OK

cansend: 6A52 41F#CC.33.F0.F0.CC.33.F0.F0
spartan: 6A52 41F#CC.33.F0.F0.CC.33.F0.F0
FRAME 000003 OK

cansend: 07AA 14CFC3C3#F0.F0
spartan: 07AA 14CFC3C3#F0.F0
FRAME 000004 OK

cansend: 29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
spartan: 29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
FRAME 000005 OK

cansend: 400D 14CFC3C3#R8
spartan: 400D 14CFC3C3#R8
FRAME 000006 OK

cansend: 7444 720#0B
spartan: 7444 720#0B
FRAME 000007 OK

cansend: 5BC8 720#
spartan: 5BC8 720#
FRAME 000008 OK

cansend: 5110 555#R1
spartan: 5110 555#R1
FRAME 000009 OK

cansend: 5A40 41F#CC.33.F0.F0.CC.33.F0
spartan: 5A40 41F#CC.33.F0.F0.CC.33.F0
FRAME 000010 OK

cansend: 3F4D 41F#CC.33.F0.F0.CC.33
spartan: 3F4D 41F#CC.33.F0.F0.CC.33
FRAME 000011 OK

cansend: 47CF 14CFC3C3#
spartan: 47CF 14CFC3C3#
FRAME 000012 OK

cansend: 6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
spartan: 6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
FRAME 000013 OK

cansend: 67F7 14CFC3C3#R7
spartan: 67F7 14CFC3C3#R7
FRAME 000014 OK

cansend: 4392 720#0B.0C
spartan: 4392 720#0B.0C
FRAME 000015 OK

cansend: 438F 14CFC3C3#F0.F0.F0.F0.CC.33
spartan: 438F 14CFC3C3#F0.F0.F0.F0.CC.33
FRAME 000016 OK

cansend: 1447 720#0B.0C.0D
spartan: 1447 720#0B.0C.0D
FRAME 000017 OK

cansend: 7379 720#0B.0C.0D.0E
spartan: 7379 720#0B.0C.0D.0E
FRAME 000018 OK

Error Cnt = 000000

a
```

```
00000000001 Rx  20000 0120
00000000002 Rx  30000 0130
00000000003 Rx  38000 0140
00000000004 Rx  3C000 0150
00000000005 Rx  3E000 0160
00000000006 Rx  3F000 0170
00000000007 Rx  3F800 0180
00000000008 Rx  3FC00 0190
00000000009 Rx  3FE00 01A0
00000000010 Rx  3FF00 01B0
00000000011 Rx  3FF80 01C0
00000000012 Rx  3FFC0 01D0
00000000013 Rx  3FFE0 01E0
00000000014 Rx  3FFF0 01F0
00000000015 Rx  3FFF8 0200
00000000016 Rx  3FFFC 0210
00000000017 Rx  3FFFE 0220
00000000018 Rx  3FFFF 0230

Naechster ISR-Puffer = 18 MAX ISR-Puffer = 150
tx-int: 000000000000 rx-int: 000000000018 ex-int: 000000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00118  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 28555 3FFF2 0608E
Bu 0B130 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 3FFF1 05A40
Bu 0B140 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 3FFF1 06A52
Bu 0B150 = 0F0F0 00000 00000 00000 3C3C3 12533 3FFF8 007AA
Bu 0B160 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 3FFF8 029D2
Bu 0B170 = 00000 00000 00000 00000 3C3C3 38533 3FFF8 0400D
Bu 0B180 = 00B00 00000 00000 00000 00000 01720 3FFF4 07444
Bu 0B190 = 00000 00000 00000 00000 00000 00720 3FFF4 05BC8
Bu 0B1A0 = 00000 00000 00000 00000 00000 21555 3FFF2 05110
Bu 0B1B0 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 3FFF1 05A40
Bu 0B1C0 = 0CC33 0F0F0 0CC33 00000 00000 0641F 3FFF1 03F4D
Bu 0B1D0 = 00000 00000 00000 00000 3C3C3 10533 3FFF8 047CF
Bu 0B1E0 = 0F0F0 0F0F0 0CC33 0AA00 3C3C3 17533 3FFF8 06A4F
Bu 0B1F0 = 00000 00000 00000 00000 3C3C3 37533 3FFF8 067F7
Bu 0B200 = 00B0C 00000 00000 00000 00000 02720 3FFF4 04392
Bu 0B210 = 0F0F0 0F0F0 0CC33 00000 3C3C3 16533 3FFF8 0438F
Bu 0B220 = 00B0C 00D00 00000 00000 00000 03720 3FFF4 01447
Bu 0B230 = 00B0C 00D0E 00000 00000 00000 04720 3FFF4 07379

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:
PCAN -> SpartanMC test in replay mode - end on 2016-12-20 15:08:08    2.case
Error Cnt = 000000
```

Mit den rot markierten Informationen kann die Zugehörigkeit der jeweiligen Zeilen aus den einzelnen Teilen der *.log Datei ermittelt werden.

Datei: Rx_2016-12-20_15-08-04.log ohne Kommentare.

```
PCAN -> SpartanMC test in replay mode - run on 2016-12-20 15:08:04    2.case

cansend: 608E 555#R8
spartan: 608E 555#R8
FRAME 000001 OK

cansend: 5A40 41F#CC.33.F0.F0.CC.33.F0
spartan: 5A40 41F#CC.33.F0.F0.CC.33.F0
FRAME 000002 OK

cansend: 6A52 41F#CC.33.F0.F0.CC.33.F0.F0
spartan: 6A52 41F#CC.33.F0.F0.CC.33.F0.F0
FRAME 000003 OK

cansend: 07AA 14CFC3C3#F0.F0
spartan: 07AA 14CFC3C3#F0.F0
FRAME 000004 OK

cansend: 29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
spartan: 29D2 14CFC3C3#F0.F0.F0.F0.CC.33.AA.55
FRAME 000005 OK

cansend: 400D 14CFC3C3#R8
spartan: 400D 14CFC3C3#R8
FRAME 000006 OK

cansend: 7444 720#0B
spartan: 7444 720#0B
FRAME 000007 OK

cansend: 5BC8 720#
spartan: 5BC8 720#
FRAME 000008 OK

cansend: 5110 555#R1
spartan: 5110 555#R1
FRAME 000009 OK

cansend: 5A40 41F#CC.33.F0.F0.CC.33.F0
spartan: 5A40 41F#CC.33.F0.F0.CC.33.F0
FRAME 000010 OK

cansend: 3F4D 41F#CC.33.F0.F0.CC.33
spartan: 3F4D 41F#CC.33.F0.F0.CC.33
FRAME 000011 OK

cansend: 47CF 14CFC3C3#
spartan: 47CF 14CFC3C3#
FRAME 000012 OK

cansend: 6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
spartan: 6A4F 14CFC3C3#F0.F0.F0.F0.CC.33.AA
FRAME 000013 OK

cansend: 67F7 14CFC3C3#R7
spartan: 67F7 14CFC3C3#R7
FRAME 000014 OK

cansend: 4392 720#0B.0C
spartan: 4392 720#0B.0C
FRAME 000015 OK

cansend: 438F 14CFC3C3#F0.F0.F0.F0.CC.33
spartan: 438F 14CFC3C3#F0.F0.F0.F0.CC.33
FRAME 000016 OK

cansend: 1447 720#0B.0C.0D
spartan: 1447 720#0B.0C.0D
FRAME 000017 OK

cansend: 7379 720#0B.0C.0D.0E
spartan: 7379 720#0B.0C.0D.0E
FRAME 000018 OK

Error Cnt = 000000

a

00000000001 Rx  20000 0120
00000000002 Rx  30000 0130
00000000003 Rx  38000 0140
00000000004 Rx  3C000 0150
00000000005 Rx  3E000 0160
00000000006 Rx  3F000 0170
00000000007 Rx  3F800 0180
00000000008 Rx  3FC00 0190
00000000009 Rx  3FE00 01A0
00000000010 Rx  3FF00 01B0
00000000011 Rx  3FF80 01C0
```

```
00000000012 Rx  3FFC0 01D0
00000000013 Rx  3FFE0 01E0
00000000014 Rx  3FFF0 01F0
00000000015 Rx  3FFF8 0200
00000000016 Rx  3FFFC 0210
00000000017 Rx  3FFFE 0220
00000000018 Rx  3FFFF 0230

Naechster ISR-Puffer = 18 MAX ISR-Puffer = 150
tx-int: 00000000000 rx-int: 00000000018 ex-int: 00000000000
Tx Belegung 0x00000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00118  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 28555 3FFF2 0608E
Bu 0B130 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 3FFF1 05A40
Bu 0B140 = 0CC33 0F0F0 0CC33 0F0F0 00000 0841F 3FFF1 06A52
Bu 0B150 = 0F0F0 00000 00000 00000 3C3C3 12533 3FFF8 007AA
Bu 0B160 = 0F0F0 0F0F0 0CC33 0AA55 3C3C3 18533 3FFF8 029D2
Bu 0B170 = 00000 00000 00000 00000 3C3C3 38533 3FFF8 0400D
Bu 0B180 = 00B00 00000 00000 00000 00000 01720 3FFF4 07444
Bu 0B190 = 00000 00000 00000 00000 00000 00720 3FFF4 05BC8
Bu 0B1A0 = 00000 00000 00000 00000 00000 21555 3FFF2 05110
Bu 0B1B0 = 0CC33 0F0F0 0CC33 0F000 00000 0741F 3FFF1 05A40
Bu 0B1C0 = 0CC33 0F0F0 0CC33 00000 00000 0641F 3FFF1 03F4D
Bu 0B1D0 = 00000 00000 00000 00000 3C3C3 10533 3FFF8 047CF
Bu 0B1E0 = 0F0F0 0F0F0 0CC33 0AA00 3C3C3 17533 3FFF8 06A4F
Bu 0B1F0 = 00000 00000 00000 00000 3C3C3 37533 3FFF8 067F7
Bu 0B200 = 00B0C 00000 00000 00000 00000 02720 3FFF4 04392
Bu 0B210 = 0F0F0 0F0F0 0CC33 00000 3C3C3 16533 3FFF8 0438F
Bu 0B220 = 00B0C 00D00 00000 00000 00000 03720 3FFF4 01447
Bu 0B230 = 00B0C 00D0E 00000 00000 00000 04720 3FFF4 07379

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:
PCAN -> SpartanMC test in replay mode - end on 2016-12-20 15:08:08    2.case
Error Cnt = 000000
```

Datei: Tx_2016-12-21_15-48-12.log

```
SpartanMC -> PCAN test in replay mode - run on 2016-12-21 15:48:12    ascii.case

spartan: can0 000#545544
canread: can0 000#545544
FRAME 000001 OK

spartan: can0 000#5370617274616E
canread: can0 000#5370617274616E
FRAME 000002 OK

spartan: can0 000#432041204E
canread: can0 000#432041204E
FRAME 000003 OK

spartan: can0 000#726F636B7321
canread: can0 000#726F636B7321
FRAME 000004 OK

Error Cnt = 000000

a

00000000001 Tx  00000 0000
00000000002 Tx  00000 0000
00000000003 Tx  00000 0000
00000000004 Tx  00000 0000

Naechster ISR-Puffer = 4 MAX ISR-Puffer = 150
tx-int: 00000000004 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x20000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:
SpartanMC -> PCAN test in replay mode - end on 2016-12-21 15:48:18    ascii.case
Error Cnt = 000000
```

Datei: Tx_2016-12-21_15-45-32.log

```
SpartanMC -> PCAN test in replay mode - run on 2016-12-21 15:45:32    2.case

spartan: can0 555#R8
canread: can0 555#R8
FRAME 000001 OK

spartan: can0 41F#CC33F0F0CC33F0
canread: can0 41F#CC33F0F0CC33F0
FRAME 000002 OK

spartan: can0 41F#CC33F0F0CC33F0F0
canread: can0 41F#CC33F0F0CC33F0F0
FRAME 000003 OK

spartan: can0 14CFC3C3#F0F0
canread: can0 14CFC3C3#F0F0
FRAME 000004 OK

spartan: can0 14CFC3C3#F0F0F0F0CC33AA55
canread: can0 14CFC3C3#F0F0F0F0CC33AA55
FRAME 000005 OK

spartan: can0 14CFC3C3#R8
canread: can0 14CFC3C3#R8
FRAME 000006 OK

spartan: can0 720#0B
canread: can0 720#0B
FRAME 000007 OK

spartan: can0 720#
canread: can0 720#
FRAME 000008 OK

spartan: can0 555#R1
canread: can0 555#R1
FRAME 000009 OK

spartan: can0 41F#CC33F0F0CC33F0
canread: can0 41F#CC33F0F0CC33F0
FRAME 000010 OK

spartan: can0 41F#CC33F0F0CC33
canread: can0 41F#CC33F0F0CC33
FRAME 000011 OK

spartan: can0 14CFC3C3#
canread: can0 14CFC3C3#
FRAME 000012 OK

spartan: can0 14CFC3C3#F0F0F0F0CC33AA
canread: can0 14CFC3C3#F0F0F0F0CC33AA
FRAME 000013 OK

spartan: can0 14CFC3C3#R7
canread: can0 14CFC3C3#R7
FRAME 000014 OK

spartan: can0 720#0B0C
canread: can0 720#0B0C
FRAME 000015 OK

spartan: can0 14CFC3C3#F0F0F0F0CC33
canread: can0 14CFC3C3#F0F0F0F0CC33
FRAME 000016 OK

spartan: can0 720#0B0C0D
canread: can0 720#0B0C0D
FRAME 000017 OK

spartan: can0 720#0B0C0D0E
canread: can0 720#0B0C0D0E
FRAME 000018 OK

Error Cnt = 000000

a

00000000001 Tx  00000 0000
00000000002 Tx  00000 0000
00000000003 Tx  00000 0000
00000000004 Tx  00000 0000
00000000005 Tx  00000 0000
00000000006 Tx  00000 0000
00000000007 Tx  00000 0000
00000000008 Tx  00000 0000
00000000009 Tx  00000 0000
00000000010 Tx  00000 0000
00000000011 Tx  00000 0000
```

```
00000000012 Tx  00000 0000
00000000013 Tx  00000 0000
00000000014 Tx  00000 0000
00000000015 Tx  00000 0000
00000000016 Tx  00000 0000
00000000017 Tx  00000 0000
00000000018 Tx  00000 0000

Naechster ISR-Puffer = 18 MAX ISR-Puffer = 150
tx-int: 00000000018 rx-int: 00000000000 ex-int: 00000000000
Tx Belegung 0x20000 Rx Belegung 0x00000 exstat: 00000 00000

CAN-work     = 00070  CAN-config   = 00001  CAN-timing   = 00151
CAN-err-cnt  = 00000  CAN-warning  = 00060  CAN-acf-sel  = 00000
CAN-acf-dat1 = 00533  CAN-acf-msk1 = 3F000  CAN-acf-dat2 = 3C3C3
CAN-acf-msk2 = 00000  CAN-err-cod  = 00000  CAN-tx-succ  = 00000
CAN-rx-succ  = 00090  CAN-tx-buff  = 00000  CAN-rx-buff  = 00000

Bu 0B120 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B130 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B140 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B150 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B160 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B170 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B180 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B190 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1A0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1B0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1C0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1D0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1E0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B1F0 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B200 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B210 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B220 = 00000 00000 00000 00000 00000 00000 00000 00000
Bu 0B230 = 00000 00000 00000 00000 00000 00000 00000 00000

Moegliche Eingaben:
id#data - Tx Frame eingeben (beginnt immer mit einer Ziffer)
a - Anzeige der Interrupts, Rx-Puffer und CAN-Register
c - Loeschen der Rx Puffer
m - CAN-Mode aendern
r - Neustart des Programms
t - Print Tx Puffer
P - Print all on
p - Print all off
D - DEBUG on
d - DEBUG off
Eingabe:
SpartanMC -> PCAN test in replay mode - end on 2016-12-21 15:45:37    2.case
Error Cnt = 000000
```