ANavS GmbH • Gotthardstraße 40 • 80686 München
www.anavs.de
Michael Heinrich

Dynamic CAN Decoder HOW TO

The Dynamic CAN Decoder allows you to use the ANavS MSRTK-System with CAN signals from your .DBC file without sharing the file with third parties or ANavS. The .dbc file used in this HOW TO is appended on the last page.

Minimal required software versions:
MAINTENANCE_TOOL=4.1.3059
DRIVER=4.1. 1807

(Both released on 23.01.2020 via beta channel)

Overview

Copy the DBC-file

Use an FTP/SCP client of your choice to copy your .DBC file to the MSRTK-System. To speed up all steps, You may want to remove unnecessary signals if your file is very long.
Let's assume you put your file to /home/pi/golf.dbc
The example file used here is included in the appendix.

Load the file

For every remaining step, you need to have access to your MSRTK-System via SSH.
On the MSRTK-System's shell, load the file with:

```
MSRTKF CAN.loadDbc -source /home/pi/golf.dbc
```

Generate code with DBCC

Now, we will run dbcc to generate the source code required to understand signals from your dbc. Though there is no hard limit on the number of signals in your DBC file, the resulting code may get very inefficient, if your file consists of millions of signals. A few thousand signals should be fine.
On the MSRTK-System's shell, run dbcc with:

```
MSRTKF CAN.dbcc
```

Select signals for sensors

Now you need to map CAN signals to sensors supported by the MSRTK-System.
Supported sensors are:

- Wheel odometry
  - Front Left (FL)
  - Front Right (FR)
  - Rear Left (RL)
  - Rear Right (RR)
- Steering angle

As of January 2020, the positioning accuracy only benefits from RL and RR.
Some cars use unsigned odometry signals and provide turning direction as a separate signal.

To view the current signal map, run on the MSRTK-System's shell:

```
MSRTKF CAN.signal -get ALL
```

To view all imported signals, run on the MSRTK-System's shell:

```
MSRTKF CAN.signal
```

In our example, we see these:

```
>> MSRTKF CAN.signal
can_0x216_WheelDirection.WheelDirRearLeft
can_0x216_WheelDirection.WheelDirRearRight
can_0x217_WheelSpeed.WheelSpeedRearLeft
can_0x217_WheelSpeed.WheelSpeedRearRight
```

To find a specific signal, run on the MSRTK-System's shell:

```
MSRTKF CAN.signal -find SEARCHTERM
```

For example:

```
>> MSRTKF CAN.signal -find rearleft
can_0x216_WheelDirection.WheelDirRearLeft
can_0x217_WheelSpeed.WheelSpeedRearLeft
```

To map a signal to a sensor, we use the '-map' parameter. It accepts a signal for the magnitude (this may be signed or unsigned) and an optional 'sign' parameter for the turning direction. We run on the MSRTK-System's shell:

```
MSRTKF CAN.signal -map RL -magn speedrearleft -sign dirrearleft,+=0,-=1
```

The '-magn' parameter accepts a search term for the magnitude of the signal. It does not need to be the precise identifier, as long as the search turns up only on result.
The (optional) '-sign' parameter requires a search term and the enum values for forward and backward, separated by comma without space.
Here, +=1 means that a signal with value '1' announces positive wheel speed. Likewise, -=2 means that a signal with value '2' announces negative wheel speed.

We do the same thing for the rear right (RR) sensor:

```
MSRTKF CAN.signal -map RR -magn speedrearright -sign dirrearright,+=0,-=1
```

If we made a mistake, we can delete a configuration:

```
MSRTKF CAN.signal -clear FL
```

Finally, we review our complete setup:

```
MSRTKF CAN.signal -get ALL
FL magnitude: null
FL signum: null
FR magnitude: null
FR signum: null
RL magnitude: can_0x217_WheelSpeed.WheelSpeedRearLeft
RL signum: can_0x216_WheelDirection.WheelDirRearLeft,+=0,-=1
RR magnitude: can_0x217_WheelSpeed.WheelSpeedRearRight
RR signum: can_0x216_WheelDirection.WheelDirRearRight,+=0,-=1
STEER magnitude: null
STEER signum: null


Signals are valid (null counts as disabled)
```

Note that you should only map signals that are present on your CAN bus,
as the system waits for all mapped signals to arrive for a complete measurement.

Tel: +49 89 89056721
Fax: +49 89 89056720
Mail: info@anavs.de

Sitz der Gesellschaft: München
Handelsregister: München HRB 195425
Umsatzsteuer ID: DE280713721

Geschäftsführung
Dr.-Ing. Patrick Henkel
Dipl.-Ing. Peter Schmitz

Commerzbank
IBAN: DE80 7004 0048 0794 7005 00
BIC: COBADEFFXXX

Compile Generated Code

In this step, all generated code get's compiled and installed on the system. Once it is installed, it will remain fully configured on the system through reboots and software updates. It will take effect as soon as the process completes.

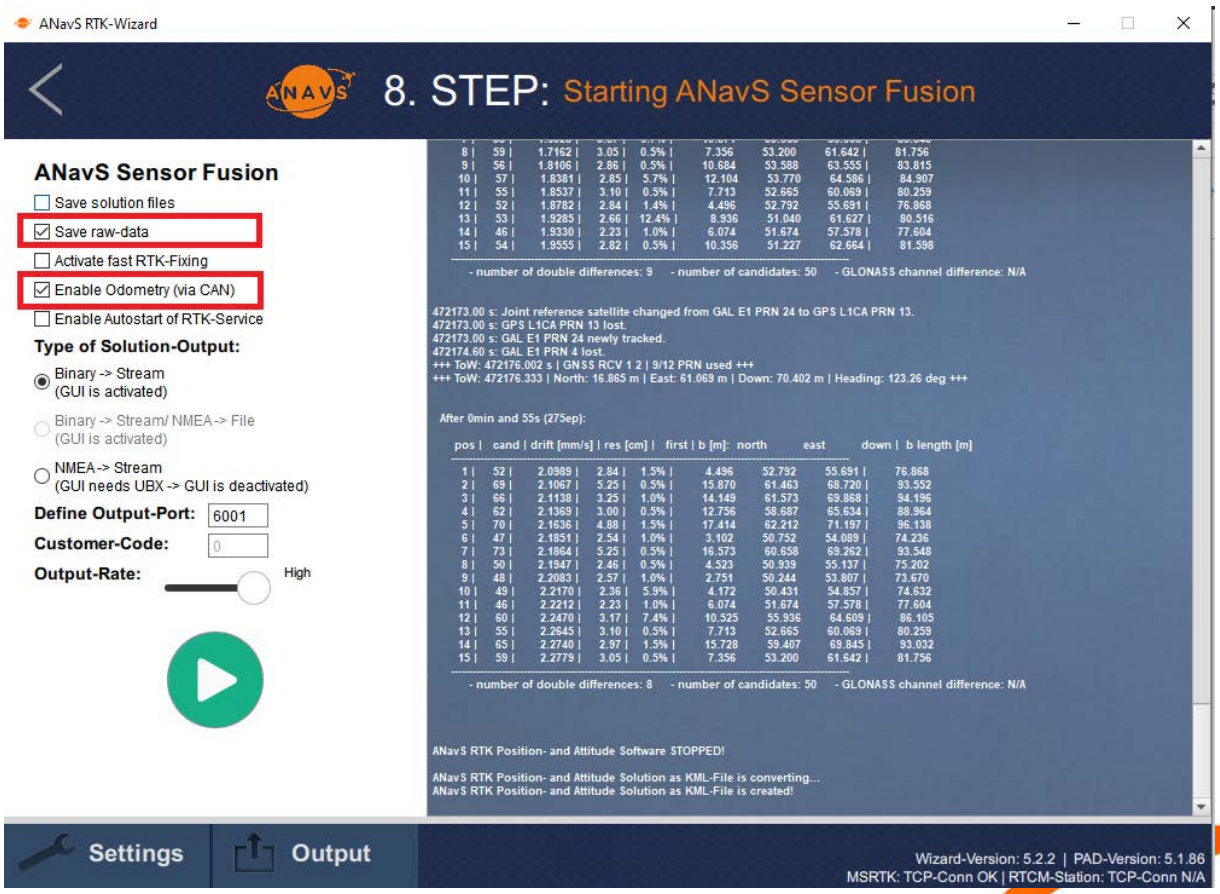
To have the configuration cross-checked again, build and install the decoder, run:

```
MSRTKF CAN.generateDecoder
```

You may get some warnings depending on your .dbc file and signal map. If everything worked, it will say: "Decoder built […]" in the end. The driver will restart immediately and load the new decoder.

Starting Sensor fusion with Wizard

The last step is enabling the CAN-Interface and saving raw data for post-processing. Please see the figure below which checkboxes (signed in red) must be activated in the Wizard.



ANavS GmbH • Gotthardstraße 40 • 80686 München
www.anavs.de
Michael Heinrich

Appendix

If you are concerned about undisclosed information in your DBC-file, you can remove most traces of it from the MSRTK-System without impeding operation of the CAN functionality.
Files that contain information on your CAN signals after the setup process:

Path	Description	CAN-related content	Clean up notes
/home/pi/device/can_profile.dbc	Copy of your .dbc file	Your DBC	File may be deleted, not required for operation
/home/pi/device/settings	Database for MSRTK-device specific settings	Your signal map	Remove CAN related lines manually, not required for operation
/home/pi/.generator/	Working directory of build process	Generated source code of decoder, binaries	Directory may be deleted, not required for operation
/share/driver/dbcc/can_profile.dbc	Copy of your .dbc file	Your DBC	File may be deleted, not required for operation
/share/driver/dbcc/can_profile.c	Source code of generated decoder	Extensive information on your signals	File may be deleted, not required for operation
/share/driver/dbcc/can_profile.h	Header of generated decoder	Some Information on your signals	File may be deleted, not required for operation
/home/pi/device/libdynamic-decoder.so	Installed decoder binaries	Executable decoder + names of signals	**Required for operation**

ANavS GmbH • Gotthardstraße 40 • 80686 München
www.anavs.de
Michael Heinrich

Example golf.dbc file:

```
VERSION ""


NS_ :
	NS_DESC_
	CM_
	BA_DEF_
	BA_
	VAL_
	CAT_DEF_
	CAT_
	FILTER
	BA_DEF_DEF_
	EV_DATA_
	ENVVAR_DATA_
	SGTYPE_
	SGTYPE_VAL_
	BA_DEF_SGTYPE_
	BA_SGTYPE_
	SIG_TYPE_REF_
	VAL_TABLE_
	SIG_GROUP_
	SIG_VALTYPE_
	SIGTYPE_VALTYPE_

BS_:

BU_:

BO_ 535 WheelSpeed: 8 Car
 SG_ WheelSpeedRearLeft : 33|15@1+ (0.0033,0) [0|0] "" Car
 SG_ WheelSpeedRearRight : 49|15@1+ (0.0033,0) [0|0] "" Car

BO_ 534 WheelDirection: 8 Car
 SG_ WheelDirRearLeft : 33|1@1+ (1,0) [0|0] "" Car
 SG_ WheelDirRearRight : 39|1@1+ (1,0) [0|0] "" Car
```


Tel: +49 89 89056721
Fax: +49 89 89056720
Mail: info@anavs.de

Sitz der Gesellschaft: München
Handelsregister: München HRB 195425
Umsatzsteuer ID: DE280713721

Geschäftsführung
Dr.-Ing. Patrick Henkel
Dipl.-Ing. Peter Schmitz

Commerzbank
IBAN: DE80 7004 0048 0794 7005 00
BIC: COBADEFFXXX