

## AN100738-000: Setting Up User specific Communication Commands

### Introduction

This document is based on Application Note 100631  
>Customizing KannMOTION drives, w. ANSI C-code>

This document shall enable you w. an example, how to implement a small smart communication Code set in ANSI-C, to get your own special functionality over an RS232 or RS485.



### How it works

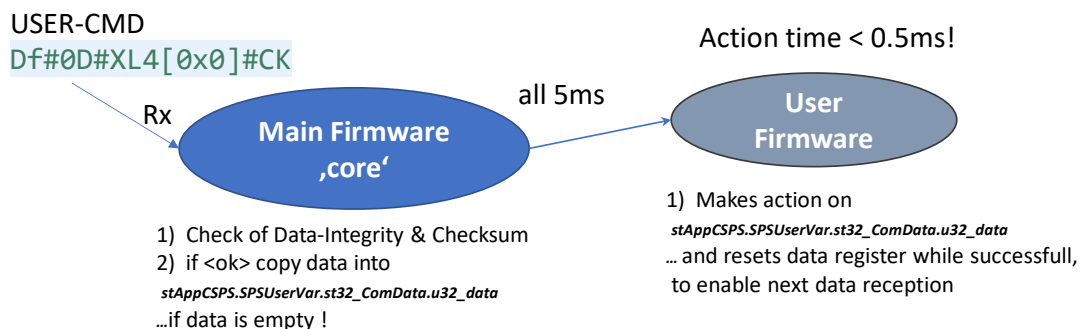
There is a base firmware running on our KannMOTION drivers, this firmware handles, communication, motor driving, positioning, error-management, io-scanning, .... means all base features of our drives.

As a very special extension, this base (core) firmware calls user code in user code section, if Checksum and function pointers are valid.

At KannMotion drives including firmware Release > V2.5-000 there is also a possibility to implement USER specific communication commands. Command Head is <Df> followed as usually by a carriage return, 4-databytes and the Checksum.

The core firmware handles this command as usual, so, means checks the signal integrity, send the echo, and if data register is free and command is ok, sets the received data into the user-rx register.

**Picture 1: basic call principle, data receive**



The drivers firmware principle is based on 'cooperative multitasking', what means, that no task is allowed to block the operation, big working load needs to be splitted in smaller working packages !

### Application code

You may download a sample code, this code is showing how the User communication might be used, and also how user specific answers can be sent.

Download of Demo- C-Files:

[http://kannmotion.adlos.com/download/ANs/AN100738\\_SampleCode.zip](http://kannmotion.adlos.com/download/ANs/AN100738_SampleCode.zip)

```

/*****/
/*!
 * \brief      SPS-USER Funktion / Demo <4>
 * \details    Building up an OWN RS232 command interpreter on <Df> instruction
 *             This DEMO needs at least Firmware Version V2.5-000 and KannMOTION-Manager 1.5.x.x
 *
 * -> we use First Byte of Df Instuction as Command, 2..4 Byte as Extra Data
 * CMD [0x00]: nothing
 * CMD [0x01]: GotoValvePosition: DataRange [0..5]      e.g. GotoPos 5      <Df#0D#XL1[0x01]#XL3[5]#CK>
 * CMD [0x40]: StoreValvePosOffset: Data 24Bit signed  e.g. OffsetPos 21000 <Df#0D#XL1[0x40]#XL3[21000]#CK>
 * CMD [0x80]: GetActualValvePosition: NoData          e.g. GetValvePos   <Df#0D#XL1[0x80]#XL3[0]#CK>
 * CMD [0x81]: ChangeGotoPos Max-Speed / Data 1..200  in HalfPercent of MaxSpeed Param e.g. Comwatch
 *                                                     Terminal: <Df#0D#XL1[0x81]#XL1[100]#XL2[0]#CK>
 *
 * USAGE:
 *   stAppCSPS.SPSUserVar.KxxFlexUser.i32_UserVar[0]      : as Helping Position Calculation Value
 *   stAppCSPS.SPSUserVar.UserNonVolatile.i32_UserVar[0] : Valve 0-Position Offset Absolute in 0.1°,
NonVolatile
 *
 * \author      Michael Zimmermann
 * \version     1.0
 * \date        3.3.2020
 * \copyright   ADLOS
 * \test
 * \param
 * \return
 *****/

```

## Communication Command

### <Df> Drive forward Data to User Sequence

This command allows to forward data into User specific Coding segment.

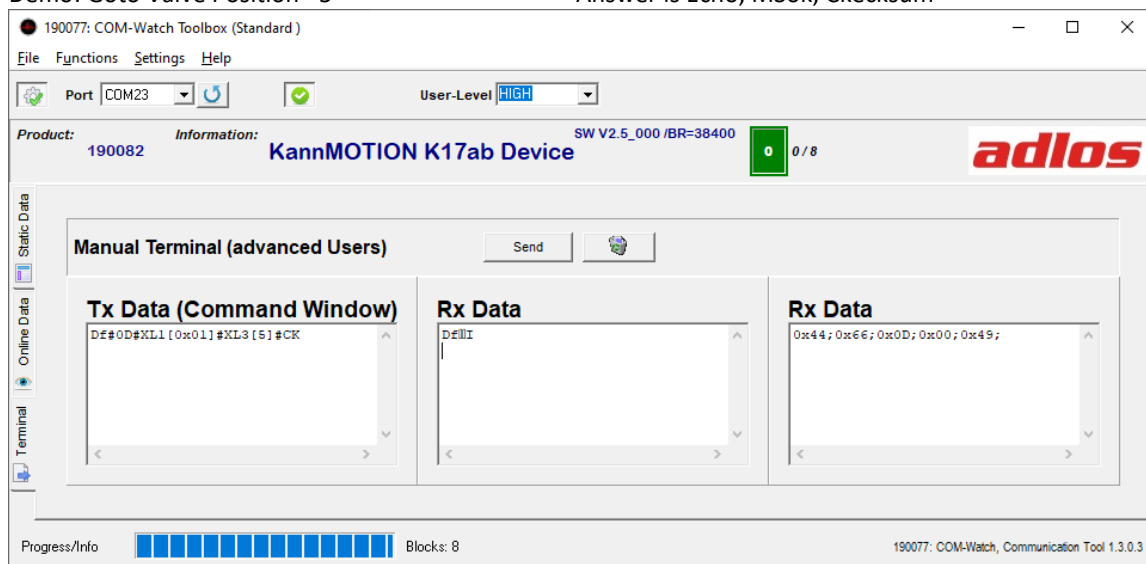
**Send:** Df<CR> <data32> <CK>  
**Answer:** Df<CR> <enDrvErrCode> <CK>  
 <data32> : forwarded data (32-Bit, interpretation User Code specifig  
 <CK> : Checksum  
 Return: enDrvErrCode see [Fehler! Verweisquelle konnte nicht gefunden werden.](#)

<Data32> ist accesible inside USER code with structure <stAppCSPS.SPSUserVar.st32\_ComData>. At the End of operation this data structure shall be set to <0> inside user code otherwise <Df> command will next time return **EMS\_ERR\_CMDnotAccepted**

## Examples with ComWatch

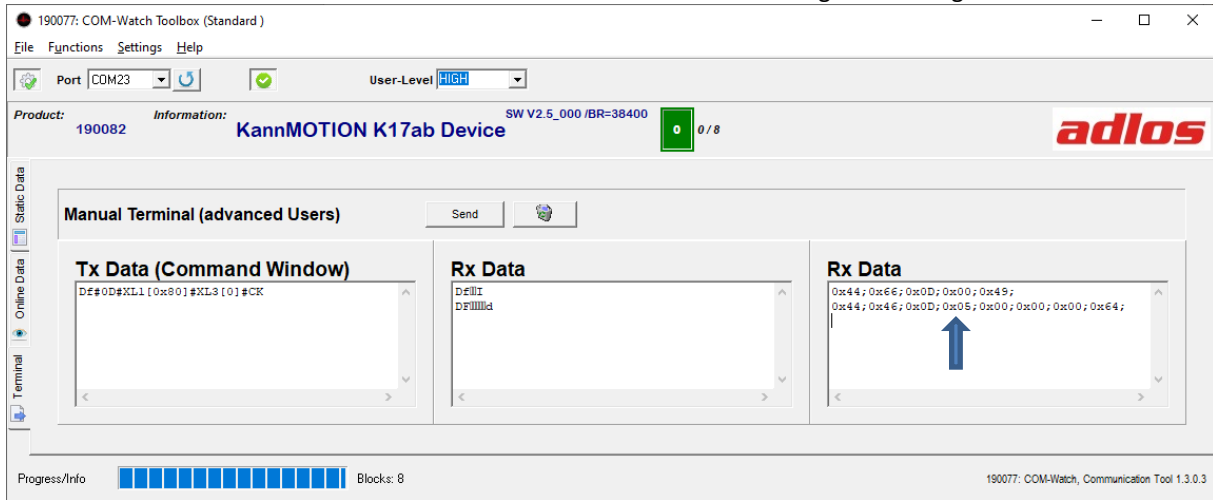
Demo: Goto Valve Position= 5

Answer is Echo, MSok, Cckecksum



Demo: GetActualValvePosition

Answer is Echo, MSok, Cchecksum  
 sec. Answer is User Message containing act Valve Position



## Downloads

Download of Demo- C-Files:

[http://kannmotion.adlos.com/download/ANs/AN100738\\_SampleCode.zip](http://kannmotion.adlos.com/download/ANs/AN100738_SampleCode.zip)

Download Communication Spec ( including Param-List):

[http://kannmotion.adlos.com/download/DOCs/100570\\_xxx\\_KannMotionProtokoll.pdf](http://kannmotion.adlos.com/download/DOCs/100570_xxx_KannMotionProtokoll.pdf)

## Tools

### Adlos Win32-APPs

adlos offers for it's customers some Helping and Design-In Tools.

#### KannMotionManager Tool (190081), manage your drives



KannMOTION Manage is the general tool for our GEN2 drives. This tool comes with an integrated C-coder and a visual drag and drop User interface for customizing your drive.

<https://kannmotion.adlos.com/download/kannmotionmanager/application/SetupKannMOTIONManager.zip>

#### ComWatch Communication Tool ( 190077 ), for Life values



ComWatch is a helping tool for engineers and technicians to explore device specific parametes, read out tracking data and settings and doing firmware updates.

The software is as it is, and in principle for free for adlos customers, the software is not made for a broad range of standard users, it's made in principle for technical engineers which are used in working w. windows based software and have some understanding of technical things.

<https://kannmotion.adlos.com/download/comwatchtool/ComWatchSetup.zip>

## Contact information

Adlos AG  
Föhrenweg 14  
FL-9496 Balzers

Thomas Vogt  
[Thomas.Vogt@adlos.com](mailto:Thomas.Vogt@adlos.com)  
Tel: +423 263 63 63

Countries: CH, A, LI, SK, IT  
[www.adlos.com](http://www.adlos.com)

KOCO MOTION GmbH  
Niedereschacher Straße 54  
D-78083 Dauchingen

Olaf Kämmerling  
[O.Kaemmerling@kocomotion.de](mailto:O.Kaemmerling@kocomotion.de)  
Tel: +49 7720/995858-0

Countries: DE, BE, NL, LU  
[www.kocomotion.de](http://www.kocomotion.de)