Manual

Agnostic Touch – Reference Manual

This reference manual describes the calibration and fine tuning of the *Agnostic Touch* modules. Make sure the *DesignStudio* is already installed on your computer. If not, please refer to the *Agnostic Touch – DevKit Quick Start Guide* for installation instructions.

Version:	2025-09-29
Company:	Algra tec AG
Author:	Niederberger Thomas

History

Change Date	Author	Description
2019-02-21	NiT	Initial version
2022-05-17	NiT	Update with protocol V2
2024-01-05	NiT	Chapter 4.11 added
2025-09-29	NiT	LED Command 0x52 removed

Content

1. Calibration procedure

The demo board is already calibrated and there is no need to recalibrate it. Nevertheless, in this chapter the calibration procedure is described step-by-step.

The basic idea of the calibration is to press each key with a force of 3 Newton (N). Based on these signals an algorithm calculates the sensitivity and the pattern for each channel. For this purpose a stick with a spring loaded pin is supplied (*3N stick*). If pushed to the mechanical stop, a force of 3 N is applied to the surface below.

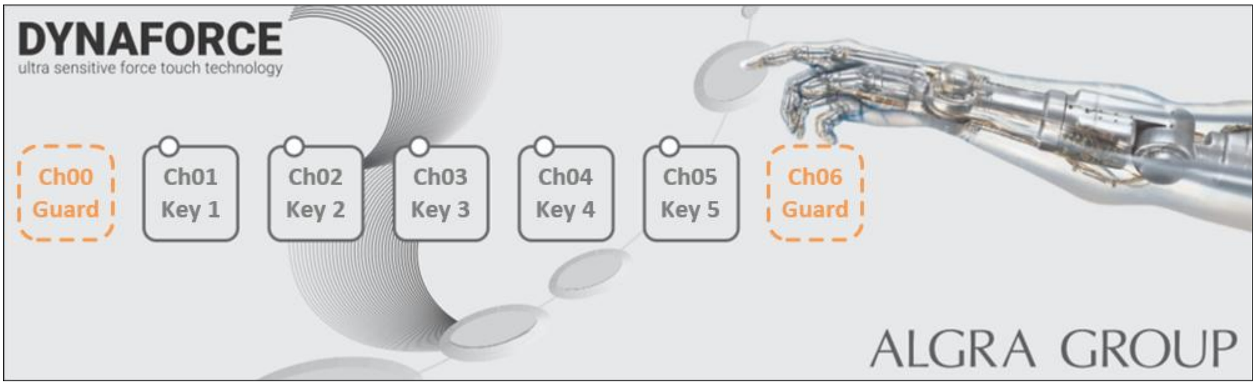
1.1 Calibration (step-by-step)

1. **Connect** your Agnostic demo board and start *DesignStudio*
2. Select the ***Channel diagram*** view.
3. **Click on *Read RawData*** to start uploading the raw data.
4. **Press with the *3N stick*** on every key including the guards (see below).
5. **Stop *Read RawData*** by releasing the button.
6. **Click on the button *Calibration*** for executing the calibration process.
7. **Click on the button *Download*** for downloading the parameters to the board.

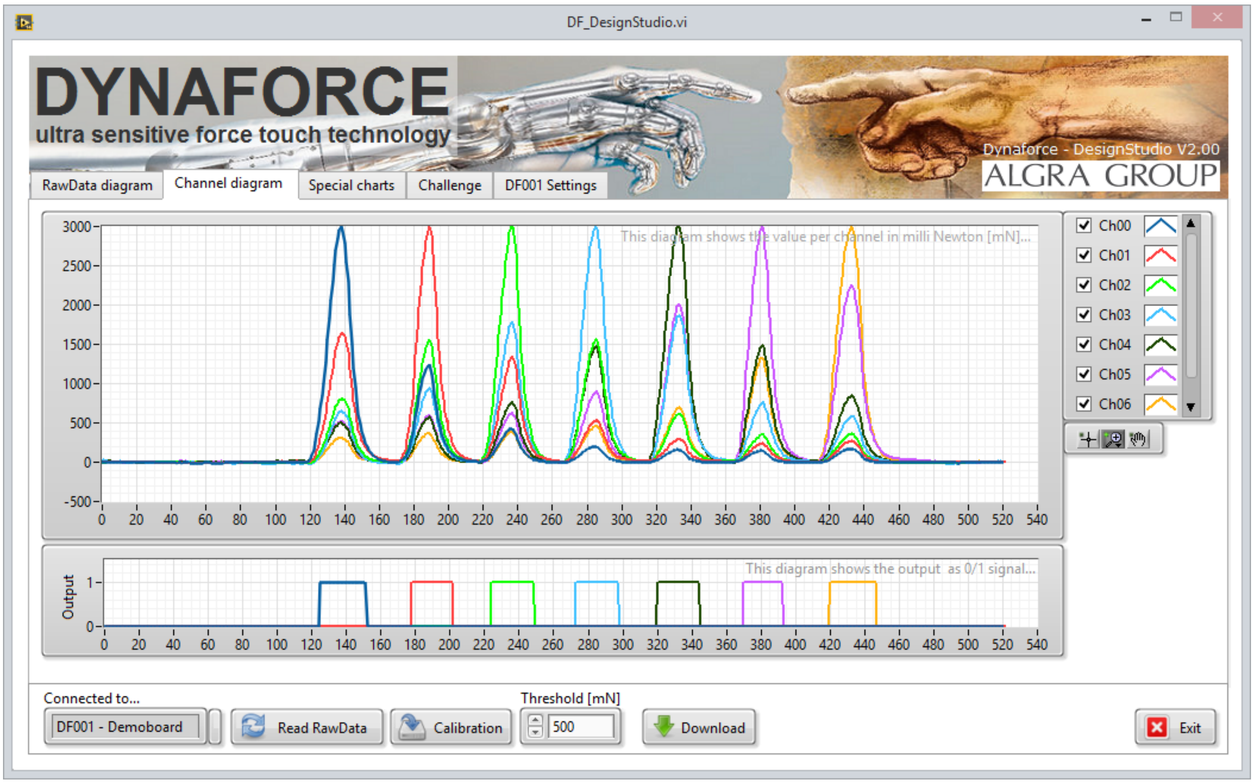
Important:
If you miscalibrated the board (e.g. you forgot to press on the guard sensors), just disconnect the board and the original parameters will be reloaded.

1.2 Guard sensors

On the demo board we use two guard sensors (Ch00, Ch06). These sensors help in defining the key press area for the nearby keys. So a strong pressure applied to the left or right of the keys can be detected by the guard sensors and filtered out by the algorithm.



1.3 Verifying the calibration

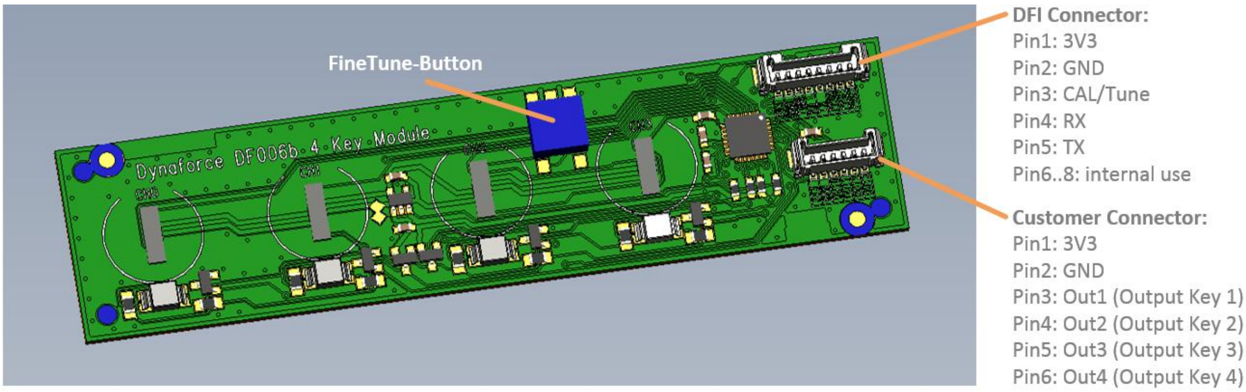


After calibration, the calibrated sensor values are visualized in the upper graph of the tab *Channel diagram*. In the lower graph, the corresponding digital output signals are plotted.

2. Using the DF006 – 4KeyModule

These 4 key modules are ready-to-use and made for a fast and easy integration of the Agnostic Touch technology behind your front panel.

The modules are already factory calibrated. But the calibration parameters may change depending on the front panel material. In this chapter we explain how to laminate and fine tune the board to your front panel material.

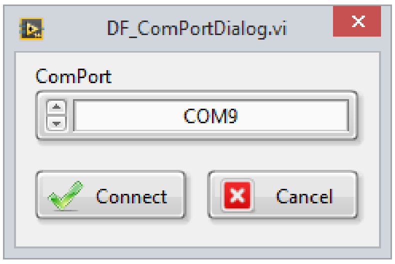


2.1 How to laminate the module behind your front panel

1. **Clean the front material using a lint-free wipe and solvent.** The mounting surface has to be flat and free of any oil residue.
2. **Peel off the liner** on the backside of the keyboard module.
3. **Stick the module** to your cleaned and dry front panel.
4. **Press** the module to the front panel. Specifically press around the sensor area (designated by the circles) to firmly laminate the module to the front panel. You may use the supplied lamination stick.

2.2 How to use the module together with *DesignStudio*

The module comes with a standard interface, called Dynaforce Interface (DFI). Use the supported DFI cable to connect the board to your computer.



Start the *DesignStudio* and choose the ComPort and the *DF006 – 4keyModule* in the dialog appearing on start up.

If you need any help, please refer to the supplied *Demo Kit - Quick Start Guide* for installation instructions and first step instructions…

After connecting the 4 key modules you have full access to the board. For example, the module can be calibrated (see above) or the threshold for the key activation force can be set.

After setting the parameters just click on *Download* to send the new parameters to the board.

Important: If you want to save the parameters permanently you have to activate the small button next to *Download* prior to downloading.

2.3 Customer Connector

Next to the standard DFI connecter on the *DF006 – 4keyModule* is a second connector. This connector is the digital interface to the customer application.

For the *DF006 – 4keyModule* we have chosen a very straight forward digital interface. The board is powered up with 3.3 VDC and a key press generates a high signal (3.3 VDC) on the corresponding output pin.

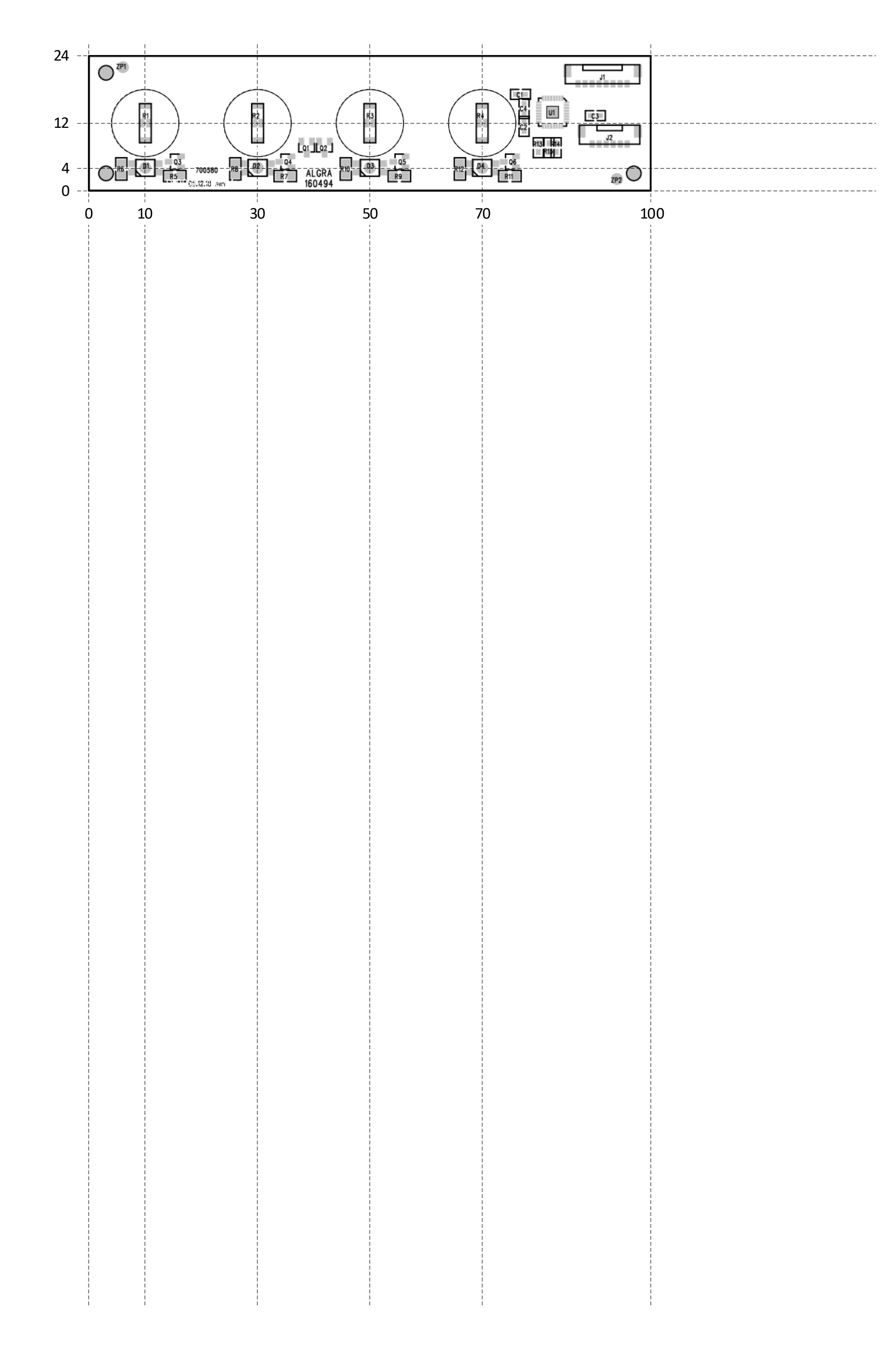
2.4 *FineTuning* behind your front without *DesignStudio*

In addition to the calibration procedure with *DesignStudio* there is a second way to fine tune your board behind your front panel without the need for *DesignStudio.* This might be useful during production in your assembly line.

The basic idea of *FineTuning* is to press each key with a *3N stick*. Based on these signals an algorithm calculates the sensitivity and the pattern for each channel according to the thickness and stiffness of your front material. The FineTuning is implemented on DF006b onwards.

FineTuning procedure (step-by-step):
1. **Power Up** the module with 3.3 Vdc by using the *DFI Connector* or the *Customer Connector.*
2. **Start FineTuning** by pushing the *FineTune-Button*. The LEDs will turn orange.
3. Wait until the orange LEDs turn off.
4. **Press each key with the *3N stick*.**
5. To end the *FineTuning* procedure, **press the *FineTune-Button* again**. If the FineTuning is successful, the LEDs turn to green. Otherwise they turn to red.

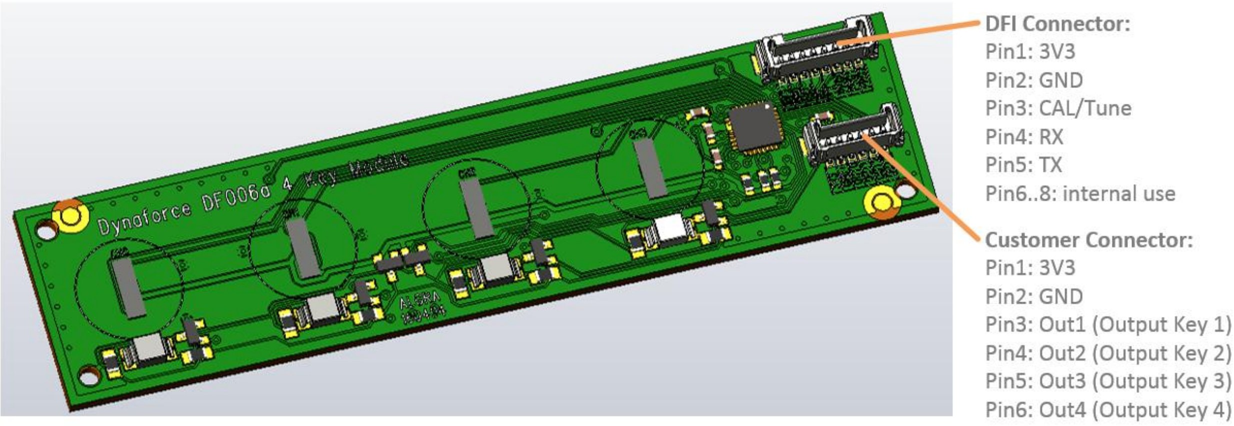
2.5 Dimensional drawing: units in mm – print in scale 100%



3. DFI – Dynaforce Interface (Protocol version V1)

Standard modules come with a serial interface. An easy way to connect the board to your computer is to use the supported DFI cable or you can directly use the RX/TX lines.

Protocol version V1 means UART (Rx/Tx) communication. V1 is a simple protocol without the possibility to address multiple boards (no address byte).



Data structure of protocol version V1:

Byte	Comment
1	Start char - 0x02
2	Lenght - Number of data bytes as ascii – i.e. 2 bytes à 0x3032
1	Command Id
0 - 255	data
1	End char - 0x03

Configuration of COM port:

Command	Data
Baudrate	57600
Data bits	8
Parity	none
Stop bit	1

The board will acknowledge a command as follow:

Command	Data
NOK	02 3030 00 03
OK	02 3030 01 03

3.1 Reading a key press

This command enables you to read the key press state for each channel as Off/On.

Request command in hex:
`02 3030 35 03`

Structure of received data:
`<STX><Length><ID><Ch00><Ch01><Ch02>…<ETX>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 8 bytes → 0x3038
<ID>	1	0x35 for this command
[Chxx]	n	Output per channel as byte – 0x00=Off, 0x01=On
<ETX>	1	End char - 0x03

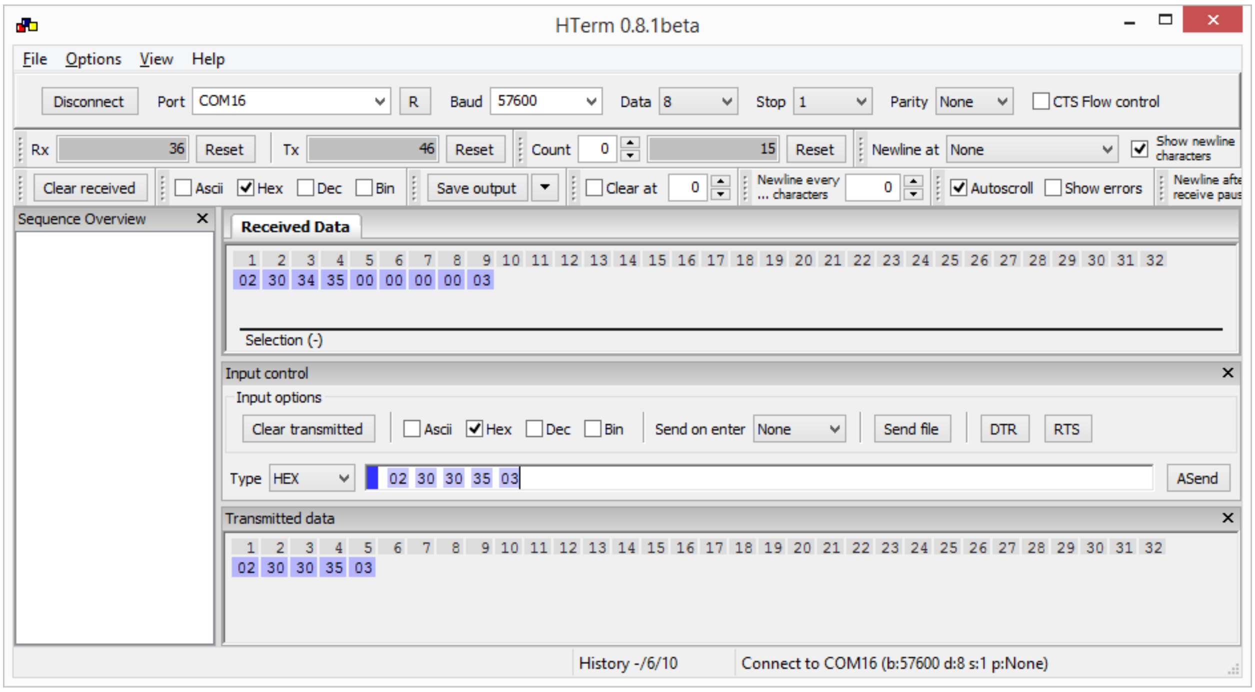
Example for DF001 - DemoBoard: 7 keys

Command	Data
02 3037 35 00 00 00 00 00 01 00 03	Ch05=On

Example for DF006 - 4keyModule: 4 keys

Command	Data
02 3034 35 00 00 01 00 03	Ch02=On

Below you find a print screen using HTerm as a terminal program and a DF006 board.
HTerm can be downloaded under https://www.heise.de/download/product/hterm-53283.



3.2 Reading force values

This command enables you to read the applied force for each channel.

Important:
This command will be ignored if the hardware doesn't support this feature.

Request command in hex:
`02 3030 33 03`

Structure of received data:
<STX><Length><ID><Ch00><Ch01><Ch02>…<ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 8 bytes → 0x3038
<ID>	1	0x33 for this command
[Chxx]	n	Value per channel as int16 – corresponds to the applied force [mN]
<ETX>	1	End char - 0x03

Example for DF006 - 4keyModule:

Command	Data
Request	02 3030 33 03
Answer	02 3038 33 0BA7 04CE 00E9 FFB2 03
Force value in [mN]	<Ch00> 0BA7 → 2983mN <Ch01> 04CE → 1230mN <Ch02> 00E9 → 233mN <Ch03> FFB2 → -78mN

3.3 Reading raw values

This command enables you to read the raw value (ADC 24bit) of each channel.
Calculation: U (ADC) [mV] = ((ADC / 2^24) * (2 * 1158 [mV])) - 1158 [mV]

Important:
This command will be ignored if the hardware doesn't support this feature.

Request command in hex:
`02 3030 30 03`

Structure of received data:
<STX><Length><ID><Ch00><Ch01><Ch02>…<ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 16 bytes → 0x3130
<ID>	1	0x30 for this command
[Chxx]	n	Value per channel as uint32
<ETX>	1	End char - 0x03

Example for DF006 - 4keyModule:

Command	Data
Request	02 3030 30 03
Answer	02 3130 30 0081 2D15 0076 5CB8 0070 DBE3 0070 6435 03
raw value (ADC 24bit)	<Ch00> 0081 2D15 → 8465685 → 10.5 mV <Ch01> 0076 5CB8 → 7756984 → -87.2 mV <Ch02> 0070 DBE3 → 7396323 → -137.0 mV <Ch03> 0070 6435 → 7365685 → -141.2 mV

3.4 Setting the LEDs

This command enables you to set the RGB value for each LED.
The command will be ignored if the hardware doesn't support this feature.

Command Structure: <STX><Length><ID><Chxx><R><G><B><ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 4 bytes → 0x3034
<ID>	1	0x53 for this command
<Chxx>	1	LED channel number as byte – Use 0xFF for setting all LEDs at once
<R>	1	0x00 to 0xFF for red LED intensity
<G>	1	0x00 to 0xFF for green LED intensity
<B>	1	0x00 to 0xFF for blue LED intensity
<ETX>	1	End char - 0x03

Example for DF001 - DemoBoard:

Command	Data
Setting LED of Ch03 to full red	02 3034 53 03 FF 00 00 03
Setting all LEDs to green	02 3034 53 FF 00 FF 00 03

Note:
- DF001/DF006 use bicolor LEDs. So, only green and red is supported.
- DF006: This functionality is implemented in FW_Version 2021-03-17 and later
- DF006: Hardware supports only LED ON/OFF – no PWM

Following command gives the LED control back to the board:
02 3031 52 01 03

3.5 Enable/disable acoustic feedback

This command enables/disables the acoustic feedback if the hardware supports this feature.

Command Structure:
<STX><Length><ID><Data><ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 1 bytes → 0x3031
<ID>	1	0x50 for this command
<Data>	1	0x01 for enable – 0x00 for disable
<ETX>	1	End char - 0x03

Example for DF001 - DemoBoard:

Command	Data
Enabling acoustic feedback	02 3031 50 01 03
Disabling acoustic feedback	02 3031 50 00 03

3.6 Enable/disable haptic feedback

This command enables/disables the haptic feedback if the hardware supports this feature.

Command Structure:
<STX><Length><ID><Data><ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 1 bytes → 0x3031
<ID>	1	0x51 for this command
<Data>	1	0x01 for enable – 0x00 for disable
<ETX>	1	End char - 0x03

Example for DF001 - DemoBoard:

Command	Data
Enabling haptic feedback	02 3031 51 01 03
Disabling haptic feedback	02 3031 51 00 03

3.7 Enable/disable optic feedback

This command enables/disables the optic feedback for all channels if the hardware supports this feature.

Command Structure:
<STX><Length><ID><Data><ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 1 bytes → 0x3031
<ID>	1	0x52 for this command
<Data>	1	0x01 for enable (LED control through the module) 0x00 for disable (LED control through serial interface)
<ETX>	1	End char - 0x03

Example for DF001 - DemoBoard:

Command	Data
Enabling optic feedback	02 3031 52 01 03
Disabling optic feedback	02 3031 52 00 03

3.8 Enable/disable key state on change

This command enables/disables the sending of the key state on change. When enabled the key state will be sent without the need of polling continuously. The key state will be sent at the rising and falling edge of the key.

Enable command in hex:
`02 3031 11 03`

Disable command in hex:
`02 3031 12 03`

The enable/disable command will be answered by:
Ok: 02 3030 01 03
Nok: 02 3030 00 03

Structure of received data after the change of key state:
<STX><Length><ID><Chxx><State><ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 2 bytes → 0x3032
<ID>	1	0x13 for this command
[Chxx]	1	Output channel number as byte
[State]	1	Key state as a byte – 0x00=Off, 0x01=On
<ETX>	1	End char - 0x03

Example:

Command	Data
02 3032 13 05 01 03	Ch05=On → signals the rising edge of channel 05
02 3032 13 05 00 03	Ch05=Off → signals the falling edge of channel 05

3.9 Reading of Serial Number

This command enables you to read the Serial Number.

Request command in hex:
`02 3030 22 03`

Structure of received data:
<STX><Length><ID><Data><ETX>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – 10 bytes → 0x3041
<ID>	1	0x22 for this command
[Data]	10	Serial Number in Ascii – Format: YYWWNNNNNN
<ETX>	1	End char - 0x03

Example:

Command	Data
Request	02 3030 22 03
Answer	02 3041 22 3231 3132 3030 3030 3935 03
Serial Number	3231 → YY → 21 (year 2021) 3132 → WW → 12 (week 12) 3030 3030 3935 → NNNNNN → 95 (current number in week)

3.10 Saving the parameters permanently

This command enables you to save the parameters permanently to the board. Without this command, the changed parameters will be lost after power down/up.

Command Structure: `<STX><Length>06<ETX>`

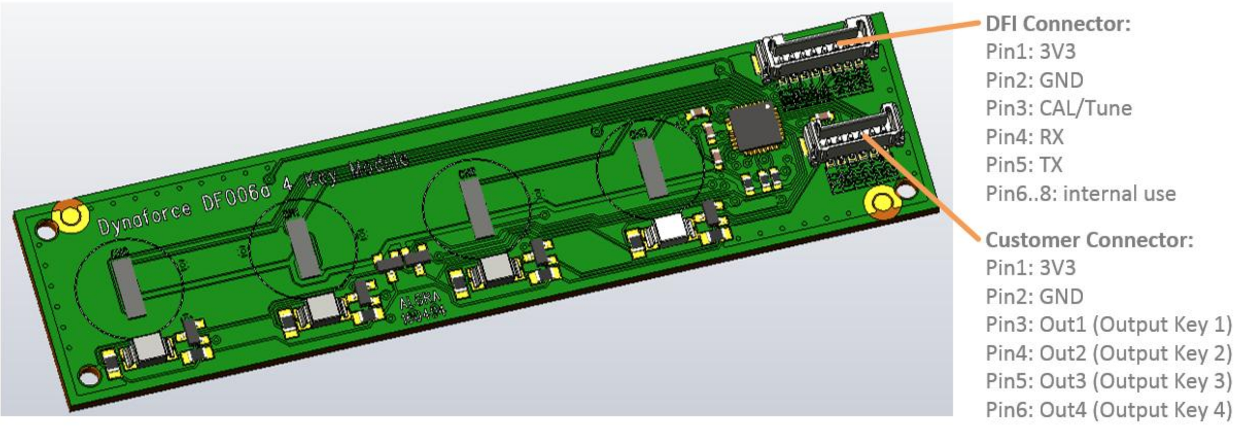
Example

Command	Data
Save	02 3030 06 03

4. DFI – Dynaforce Interface (Protocol version V2)

Many modules (i.e LeanLine boards) come with a serial interface of protocol version V2. An easy way to connect the board to your computer is to use the supported DFI cable or you can directly use the RX/TX lines.

Protocol version V2 means UART (Rx/Tx) communication. V2 is a simple protocol with the possibility to address multiple boards per address byte. Further the communication is secured by a checksum.



Data structure and checksum definition of protocol version V2:

Byte	Comment
1	Start char - 0x02
2	Lenght - Number of data bytes as ascii – i.e. 2 bytes -> 0x3032
1	BusAddress <Adr>. 0x00: Default - 0xFF: Broadcast (module answers always and with it's BusAddress)
1	Command Id
0 - 255	data
1	CheckSum <Chk> Generate Chk: Two's complement of byte addition from <Lenght> to <data> Verify Chk: byte addition from <Lenght> to <Chk> shall be zero Example: 02 3036 00 20 4446 3030 3161 FE Link: https://en.wikipedia.org/wiki/Intel_HEX

Configuration of COM port:

Command	Data
Baudrate	57600
Data bits	8
Parity	none
Stop bit	1

The board will acknowledge a command as follow:

Command	Data
NOK	02 3030 <Adr> 00 <Chk>
OK	02 3030 <Adr> 01 <Chk>

4.1 Reading the bus address

This command enables you to read the bus address of a single board. It is important that only one board is connected to the host. This command is only used if you are unsure of its board address. To get an answer from one board with an unknown bus address, the bus address 0xFF can be used.

Request command in hex:
`02 3030 FF 15 <Chk>`

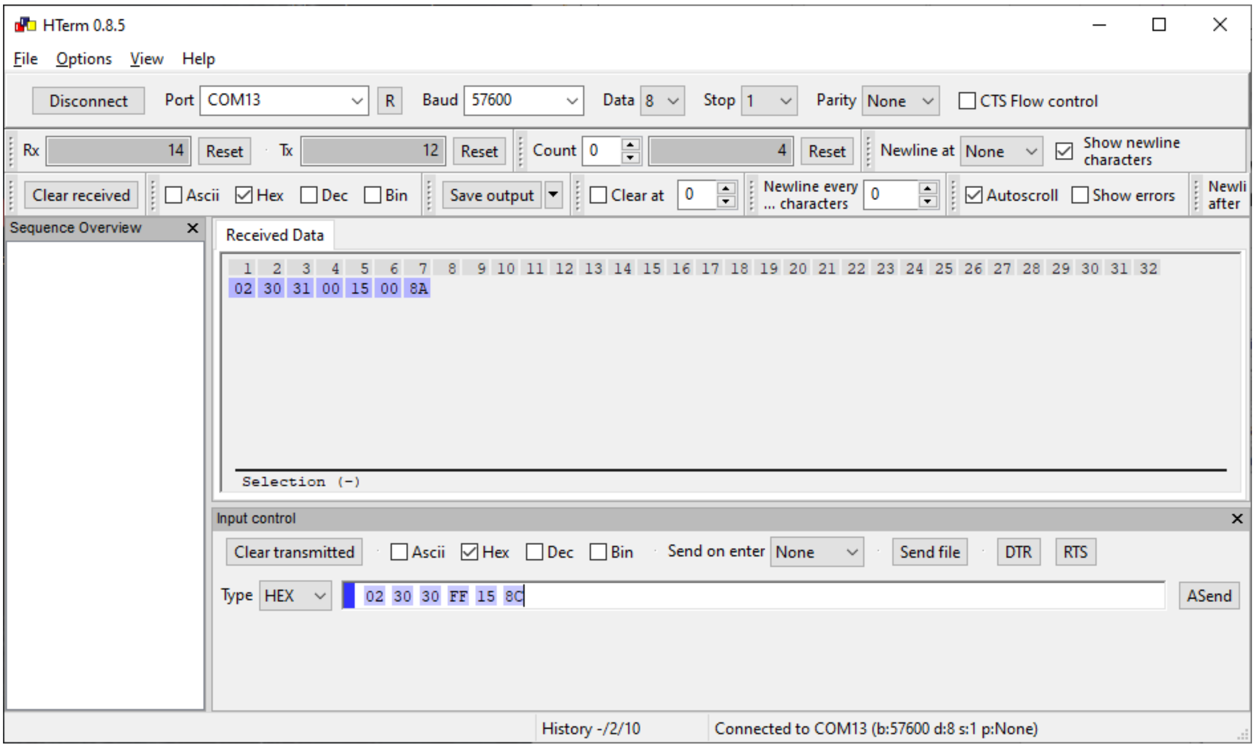
Structure of received data:
`<STX><Length><Adr><ID><Data><Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 8 bytes → 0x3038
<Adr>	1	Bus address of the connected board
<ID>	1	0x15 for this command
<Data>	1	Data – for this command data is equal to bus address
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys

Command	Data
Request	02 3030 FF 15 8C
Answer	02 3031 00 15 00 8A In this example the bus address of the connected board is 0x00.

Below you find a print screen using HTerm as a terminal program and a LL004 – LeanLine board. HTerm can be downloaded under https://www.heise.de/download/product/hterm-53283.



4.2 Setting the bus address

This command enables you to set the bus address of a board.

Command Structure: `<STX><Length><Adr><ID><NewAdr><Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 1 bytes → 0x3031
<Adr>	1	Bus address of the connected board
<ID>	1	0x15 for this command
<NewAdr>	1	New bus address for the connected board
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00

Command	Data
Setting a board with bus address 0x00 to bus address 0x0D	02 3031 00 15 0D 7D
The board answers with OK-Message and the new bus address 0x0D	02 3030 0D 01 92

Important:
With this example you set the bus address to 0x0D. For the examples below the bus address 0x00 is used. So, it's best to set the bus address back to 0x00 before proceeding.

4.3 Reading a key press

This command enables you to read the key press state for each channel as Off/On.

Request command in hex:
`02 3030 <Adr> 35 <Chk>`

Structure of received data:
`<STX><Length><Adr><ID><Ch00><Ch01><Ch02>…<Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 4 bytes → 0x3034
<Adr>	1	Bus address of the connected board
<ID>	1	0x35 for this command
[Chxx]	n	Output per channel as byte – 0x00=Off, 0x01=On
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00

Command	Data
Request	02 3030 00 35 6B
Answer	02 3034 00 35 00 00 01 00 66 Ch03=On

4.4 Reading force values

This command enables you to read the applied force for each channel.

Important:
This command will be ignored if the hardware doesn't support this feature.

Request command in hex:
`02 3030 <Adr> 33 <Chk>`

Structure of received data:
`<STX><Length><Adr><ID><Ch00><Ch01><Ch02>…<Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 8 bytes → 0x3038
<Adr>	1	Bus address of the connected board
<ID>	1	0x33 for this command
[Chxx]	n	Value per channel as int16 – corresponds to the applied force [mN]
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00

Command	Data
Request	02 3030 00 33 6D
Answer	02 3038 33 00 0BA7 04CE 00E9 FFB2 47
Force value in [mN]	<Ch00> 0BA7 → 2983mN <Ch01> 04CE → 1230mN <Ch02> 00E9 → 233mN <Ch03> FFB2 → -78mN

4.5 Reading raw values

This command enables you to read the raw value (ADC 24bit) of each channel.
Calculation: U (ADC) [mV] = ((ADC / 2^24) * (2 * 1158 [mV])) - 1158 [mV]

Important:
This command will be ignored if the hardware doesn't support this feature.

Request command in hex:
`02 3030 <Adr> 30 <Chk>`

Structure of received data:
`<STX><Length><Adr><ID><Ch00><Ch01><Ch02>…<Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 16 bytes → 0x3130
<Adr>	1	Bus address of the connected board
<ID>	1	0x30 for this command
[Chxx]	n	Value per channel as uint32
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00

Command	Data
Request	02 3030 00 30 70
Answer	02 3130 00 30 0081 2D15 0076 5CB8 0070 DBE3 0070 6435 EB
raw value (ADC 24bit)	<Ch00> 0081 2D15 → 8465685 → 10.5 mV <Ch01> 0076 5CB8 → 7756984 → -87.2 mV <Ch02> 0070 DBE3 → 7396323 → -137.0 mV <Ch03> 0070 6435 → 7365685 → -141.2 mV

4.6 Setting the LED color for the inactive State

This command enables you to set the inactive LED color.
The command will be ignored if the hardware doesn't support this feature.

Command Structure: <STX><Length><Adr><ID><Ch00><Ch01>…<Chxx><Chk>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 4 bytes → 0x3034
<Adr>	1	Bus address of the connected board
<ID>	1	0x55 for this command
<Chxx>	1	RGB value for each channel as u32 <00RRGGBB>
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00

Command	Data
Setting inactive color for all LEDs to full green	02 3100 00 55 00 00 FF 00 00 00 FF 00 00 00 FF 00 00 00 FF 00 7E
Setting inactive color for all LEDs to 10% white	02 3100 00 55 00 19 19 19 00 19 19 19 00 19 19 19 00 19 19 19 4E

4.7 Presetting the LED color for the active State

This command enables you to set the active LED color.
The command will be ignored if the hardware doesn't support this feature.

Command Structure: <STX><Length><Adr><ID><Ch00><Ch01>…<Chxx><Chk>

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 4 bytes → 0x3034
<Adr>	1	Bus address of the connected board
<ID>	1	0x54 for this command
<Chxx>	1	RGB value for each channel as u32 <00RRGGBB>
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00

Command	Data
Setting active color for all LEDs to full green	02 3100 00 54 00 00 FF 00 00 00 FF 00 00 00 FF 00 00 00 FF 00 7F
Setting active color for all LEDs to 10% white	02 3100 00 54 00 19 19 19 00 19 19 19 00 19 19 19 00 19 19 19 4F

4.8 Reading of Serial Number

This command enables you to read the Serial Number.

Request command in hex:
`02 3030 <Adr >22 <Chk>`

Structure of received data:
`<STX><Length><Adr><ID><Data><Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – 10 bytes → 0x3041
<Adr>	1	Bus address of the connected board
<ID>	1	0x22 for this command
[Data]	10	Serial Number in Ascii – Format: YYWWNNNNNN
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00:

Command	Data
Request	02 3030 00 22 7E
Answer	02 3041 00 22 3231 3132 3030 3030 3935 79
Serial Number	3231 → YY → 21 (year 2021) 3132 → WW → 12 (week 12) 3030 3030 3935 → NNNNNN → 95 (current number in week)

4.9 Saving the parameters permanently

This command enables you to save the parameters permanently to the board. Without this command, the changed parameters will be lost after power down/up.

Command Structure: `<STX><Length><Adr><ID>06<Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – i.e. 0 bytes → 0x3030
<Adr>	1	Bus address of the connected board
<ID>	1	0x06 for this command
<Chk>	1	Checksum (see above)

Example for LL004 – LeanLine board: 4 keys with bus address 0x00:

Command	Data
Save	02 3030 00 06 9A

4.10 Reading of firmware version

This command enables you to read the firmware version.

Request command in hex:
`02 3030 <Adr >21 <Chk>`

Structure of received data:
`<STX><Length><Adr><ID><Data><Chk>`

Element	Byte	Comment
<STX>	1	Start char - 0x02
<Length>	2	Number of data bytes as ascii – 10 bytes → 0x3041
<Adr>	1	Bus address of the connected board
<ID>	1	0x21 for this command
[Data]	12	Firmware version in Ascii – Format: YYMMDD
<Chk>	1	Checksum (see above)

Example with bus address 0x00:

Command	Data
Request	02 3030 00 21 7F
Answer	02 3036 00 21 3233 3039 3133 47
Firmware version	3233 → 23 (year 2023) 3039 → 09 (month 09) 3133 → 13 (day in month 13)