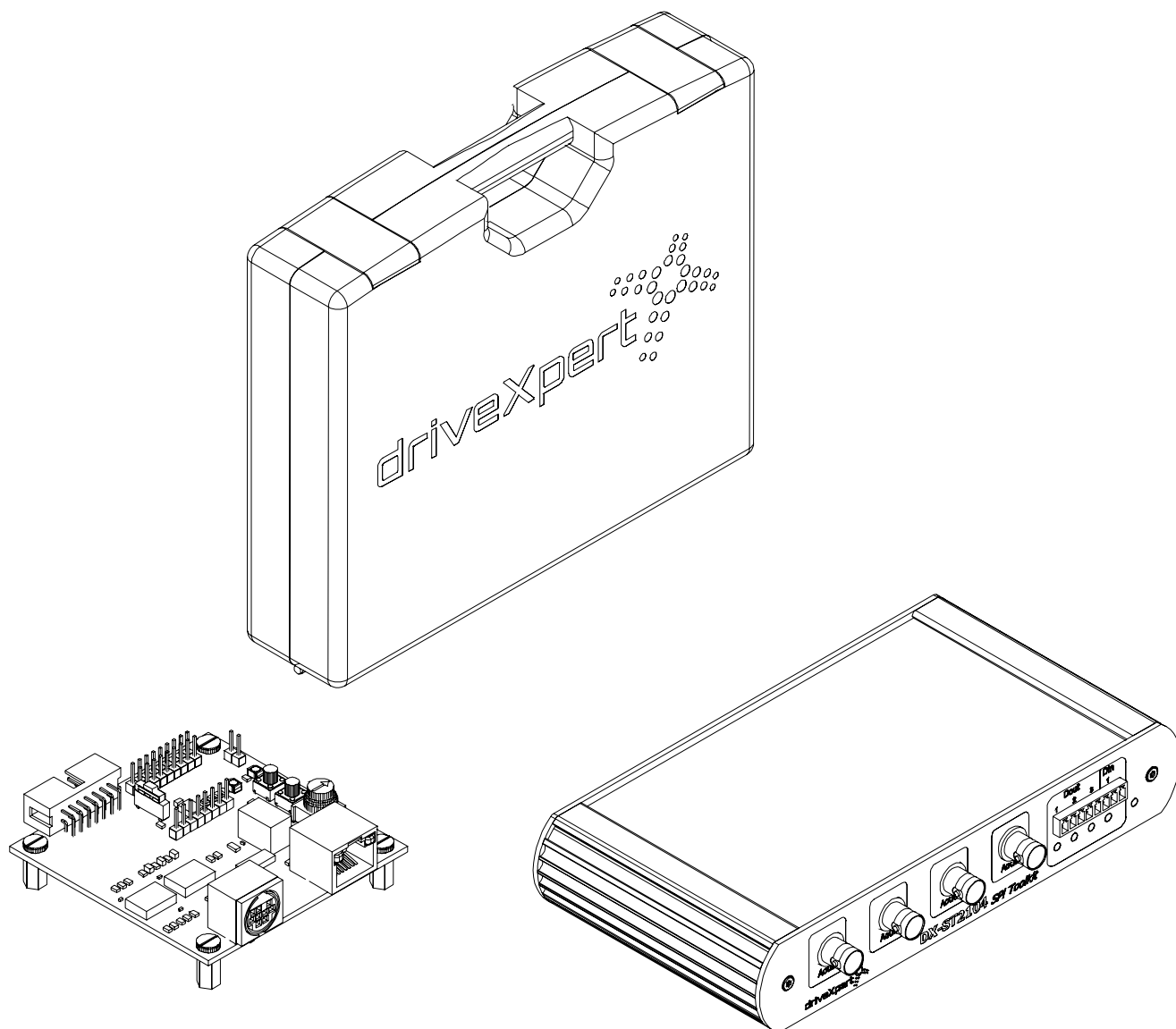


perfect drive

DX-ST2104

SPI Toolkit

User Manual



Manual structure

Chapter	Title	Content	Page
1	Components	Scope of supply description	5
2	Application scenarios	Features and possible applications for SPI Toolkits	9
3	Technical data	Electrical specifications of DX-ST2104 and additional hardware	13
4	Software installation	Installation notes for SPI Toolkit software package including data acquisition software and drivers	23
5	SPI connection establishment	Description of the method establishing a data transfer connection via SPI interface	25
6	SPIRecorder protocol	Specification of the communication protocol for data transfer between TargetBoard and DX-ST2104	31
7	Appendix	Contains schematics and pcb layouts	37

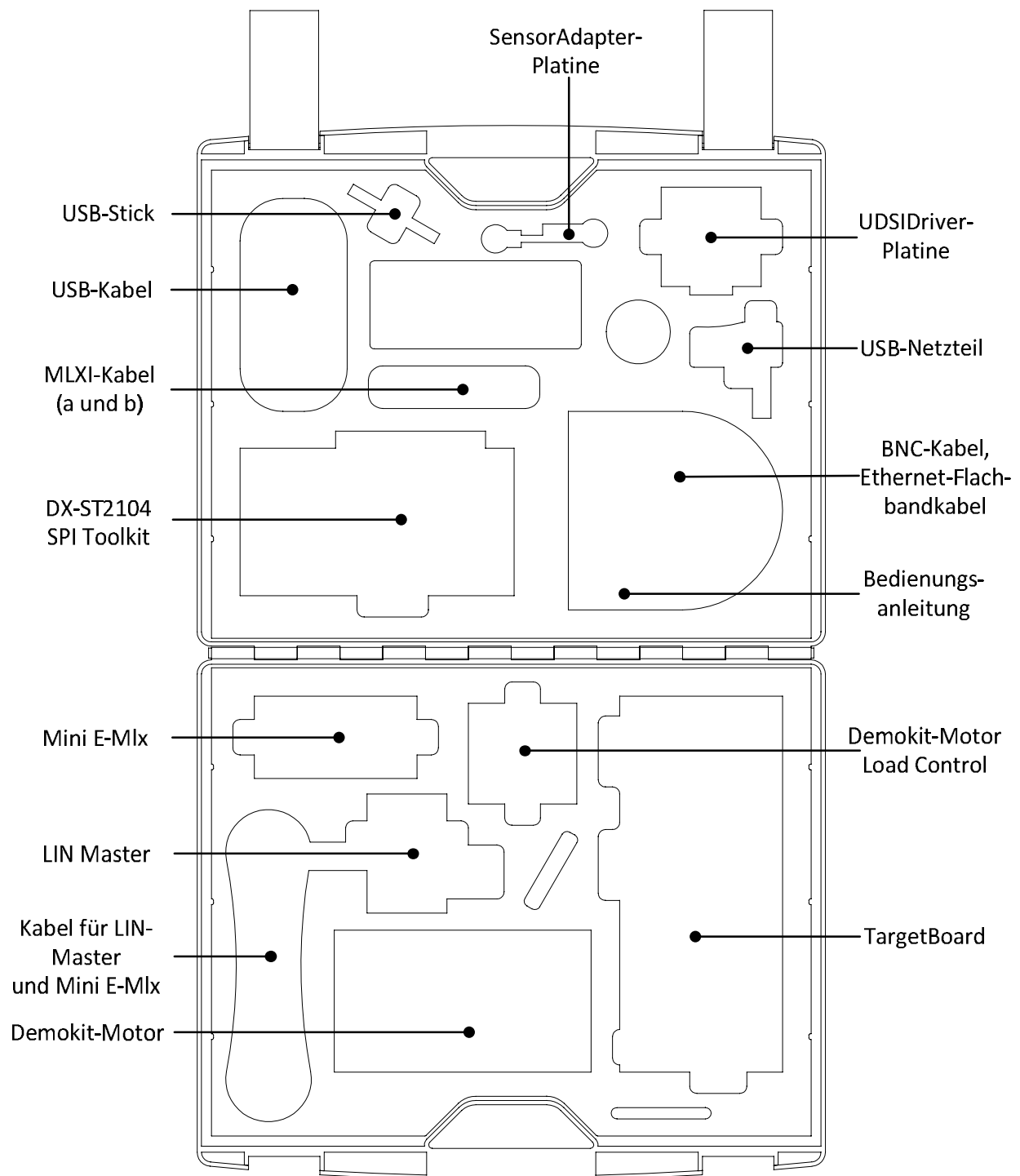
Table 0-1: Revision history

Rev.	Subject	Date	Page
0.1	Initial document	23.04.2012	x
1.0	Establishment 1st Release	24.04.2012	x
1.1	Corrected graphics	01.06.2012	x
1.2	UDSIDriver Rev 2.2 and SPI Toolkit Rev 2.3	22.03.2012	x

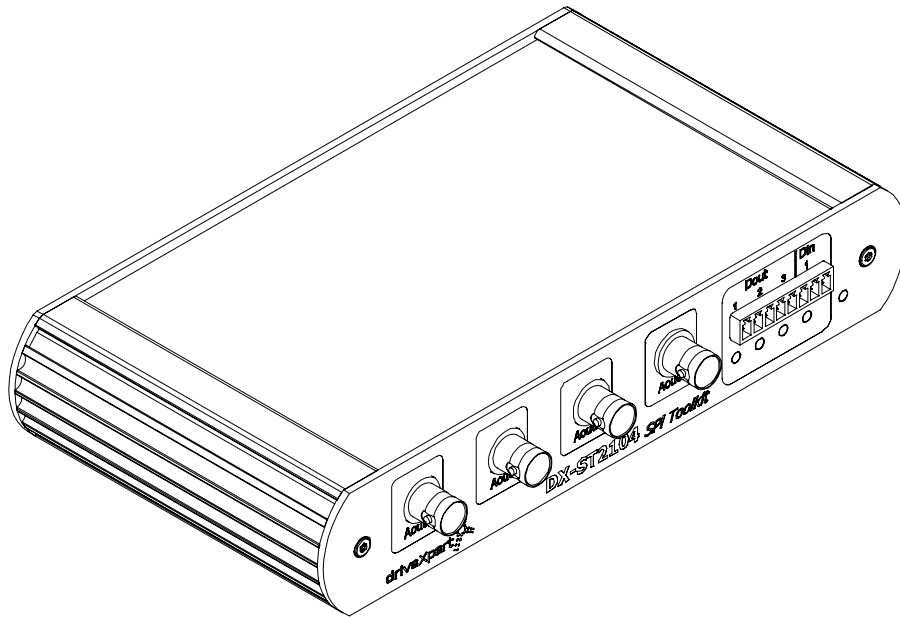
Table of content

Manual structure.....	3
1. Components.....	5
2. Application scenarios.....	9
2.1 Realtime analysis of motor control algorithms via MLXI	9
2.2 Realtime analysis of motor control algorithms via UDSI	10
2.3 Configuration of Aout channels.....	11
2.4 Supported TargetBoards / MotorController	11
2.5 Data export and integration possibilities	11
3. Technical data	13
3.1 Connectors DX-ST2104 SPI Toolkit (Front)	13
3.2 Connectors DX-ST2104 SPI Toolkit (back)	15
3.3 UDSIDriver circuit board.....	20
3.4 SensorAdapter circuit board	22
4. Software installation.....	23
4.1 DXToolkitRecorder installation.....	23
4.2 SPI Toolkit software components.....	23
4.3 System requirements	23
4.4 Software update	24
4.5 Flashing firmware.....	24
5. SPI connection establishment	25
5.1 Hardware	25
5.2 Software connection establishment with DXToolkitRecorder.....	27
5.3 Method for connection problems	28
6. SPIRecorder protocol.....	31
6.1 Protocol versions.....	31
6.2 Transmission modes and data formats	32
6.3 SPIRecorder module interface description.....	34
7. Appendix	37

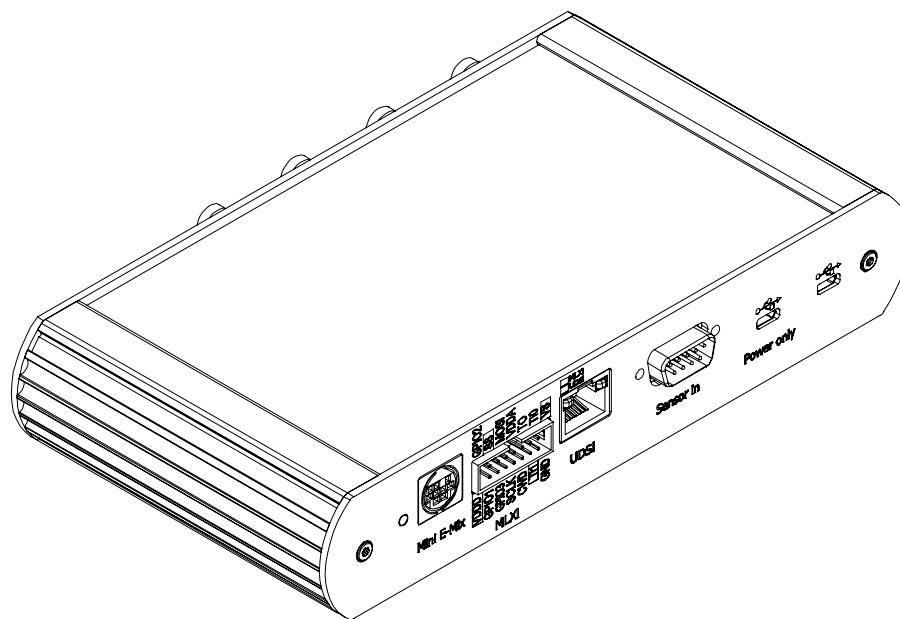
1. Components



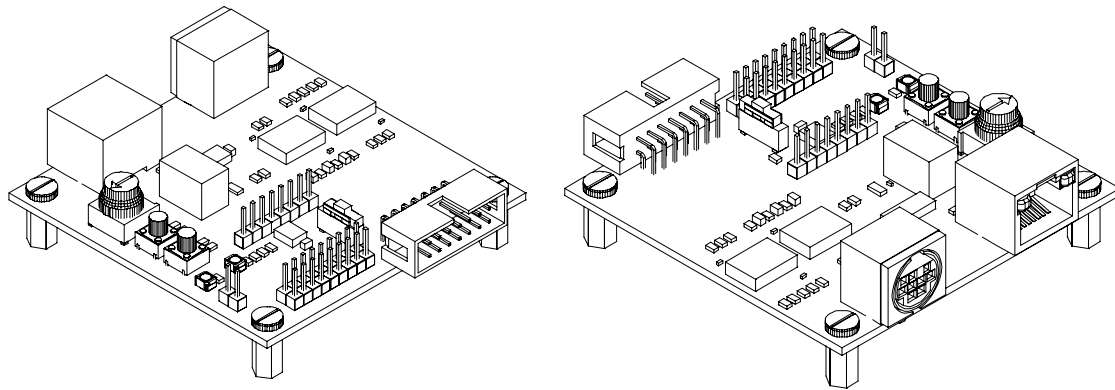
Picture 1-1: Opened SPI Toolkit case with components



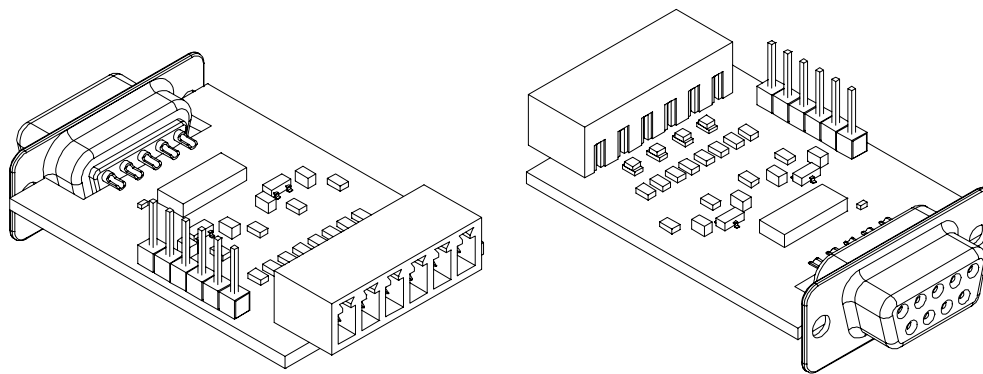
Picture 1-2: DX-ST2104 SPI Toolkit (front)



Picture 1-3: DX-ST2104 SPI Toolkit (back)



Picture 1-4: UDSIDriver circuit board (front and back)



Picture 1-5: SensorAdapter circuit board (front and back)

Table 1-1: Cables

Designation	Description	Quantity
Ethernet flat ribbon cable	Unshielded cable for connecting UDSIDriver circuit board with SPI Toolkit (RJ-45 connector)	1
Micro USB cable 1.8 m	Cable for data transmission and/or power supply	1
Micro USB cable 3 m	Cable for data transmission and/or power supply	1
BNC cable red	Connecting peripherals to Aout1 (BNC connector)	1
BNC cable yellow	Connecting peripherals to Aout2 (BNC connector)	1
BNC cable green	Connecting peripherals to Aout3 (BNC connector)	1
BNC cable blue	Connecting peripherals to Aout4 (BNC connector)	1
MLXI cable_a	With closed end	1
MLXI cable_b	With open end	1

Table 1-2: Others

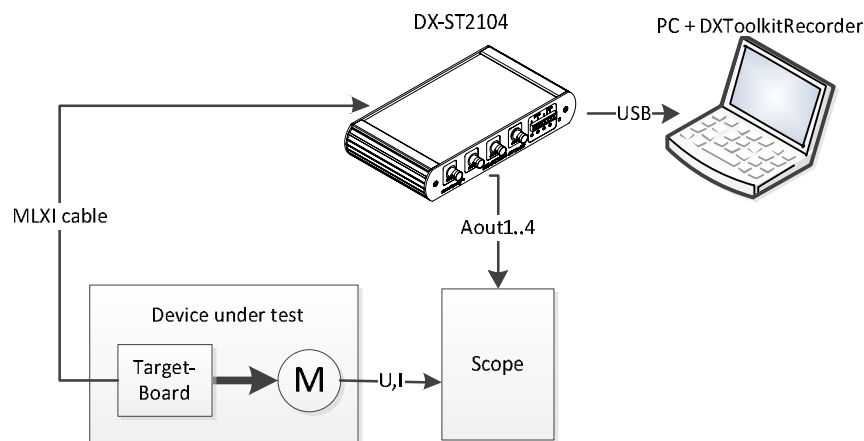
Designation	Description	Quantity
USB power supply	Power supply via USB connector	1
USB stick	Memory stick, containing necessary software	1

Table 1-3: Optional accessories (not included in standard scope of supply)

Designation	Description
Mini E-Mlx	Hardware emulator
LIN Master	LIN Master for connecting LIN Bus to a PC via USB connection
Demokit-Motor	Brushless DC Motor with brake
Demokit-Motor Load Control	Control circuit for brake, operation via LIN
TargetBoard	Experimental board, contains controller and power board to operate brushless DC Motors. For example: MLX81200 EvalBoard, MLX81200 HighPower DemoBoard, etc.

2. Application scenarios

2.1 Realtime analysis of motor control algorithms via MLXI

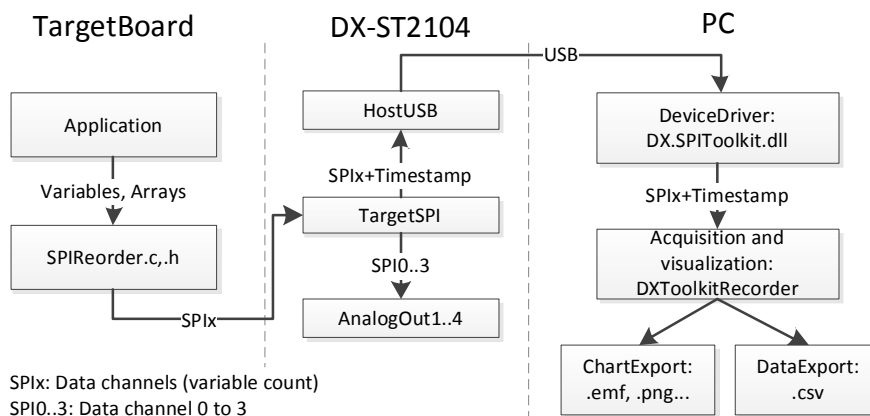


Picture 2-1: Realtime analysis of motor control algorithms via MLXI

In the most simple experimental configuration the DX-ST2104 is connected to the TargetBoard via one of the two MLXI cables. While the motor is running all incoming data and status information are transmitted to the DX-ST2104 via SPIRecorder protocol. These are the following types of data:

- ADC values
- State variables of state machines
- internal controller values (reference speed, speed error, dynamic anti wind up limits, etc.)
- all other application variables

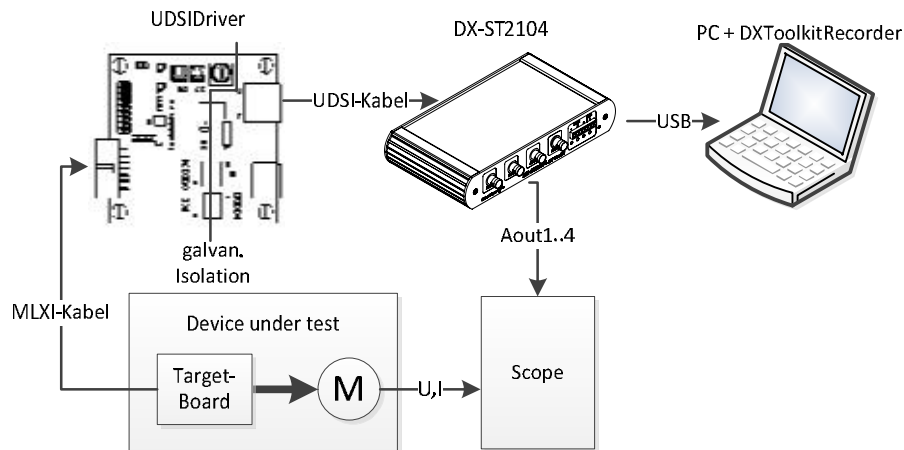
The SPI Toolkit is able to send out data analogously to an oscilloscope or another data acquisition system. In parallel every sampled data could be visualized on the DXToolkitRecorder software chart.



Picture 2-2: SPI Toolkit software components

To be able observing the interesting variables from the TargetBoard application, the program has to send variable content regularly to the SPIRecorder.c module. This module will send out all data on separate channels to the SPI Toolkit. Every information sent on SPI0 to SPI4 will be given out on Aout channels 1 to 4 and simultaneously be transmitted to the PC via USB. Higher SPI channels are only transmitted via USB.

2.2 Realtime analysis of motor control algorithms via UDSI

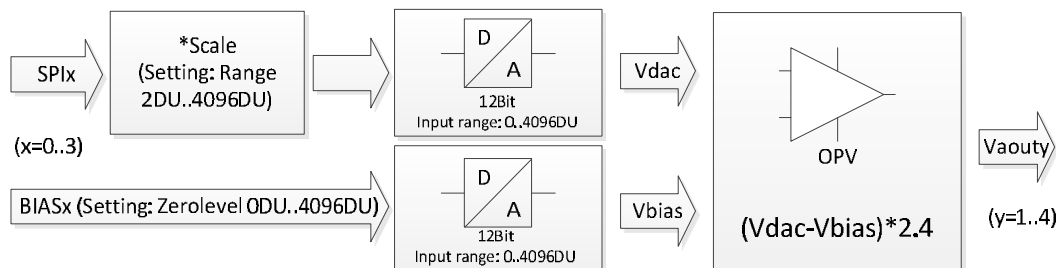


Picture 2-3: Realtime analysis of motor control algorithms via UDSI

Under testing conditions with a higher electro magnetic distortion, the communication via SPI and USB could be disturbed. In this case the direct connection DX-ST2104---DUT should be replaced by the UDSIDriver circuit board. Nevertheless the MLXI cable between the TargetBoard and the UDSIDriver circuit board has to be kept as short as possible. In contrast to this the UDSI flat ribbon cable could be even longer. SPI signals are transmitted differentially and are decoupled from the main source (the motor) of distortion.

2.3 Configuration of Aout channels

The Aout channels are four AD converters that are configurable via DXToolkitRecorder. 12Bit input values are converted to voltages between -5 V up to +5 V.



Picture 2-4: Aout channels (simplified block diagram)

In default configuration every data on SPI channel 0 up to SPI channel 3 is send out on it's corresponding Aout channel. This assignment is constant. But it is possible to send out other values to the analog output channels (for example position and velocity). The conversion is divided into three steps:

1. Scaling / RangeShift: to use the full range, set the range parameter to the expected SPI output data range
2. Analog conversion via DAC
3. Analogous BIAS compensation and amplification via an operational amplifier

The BIAS value enables an offset compensation of the digital values. At a BIAS value of about 2048 Vaout is set to 0 V if SPI data is 2048, too.

Table 2-1: Example values at a BIAS value of about 2048

SPI channel's value	Aout voltage Vaout
0	-4.92 V
2048	0 V
4096	+4.92 V

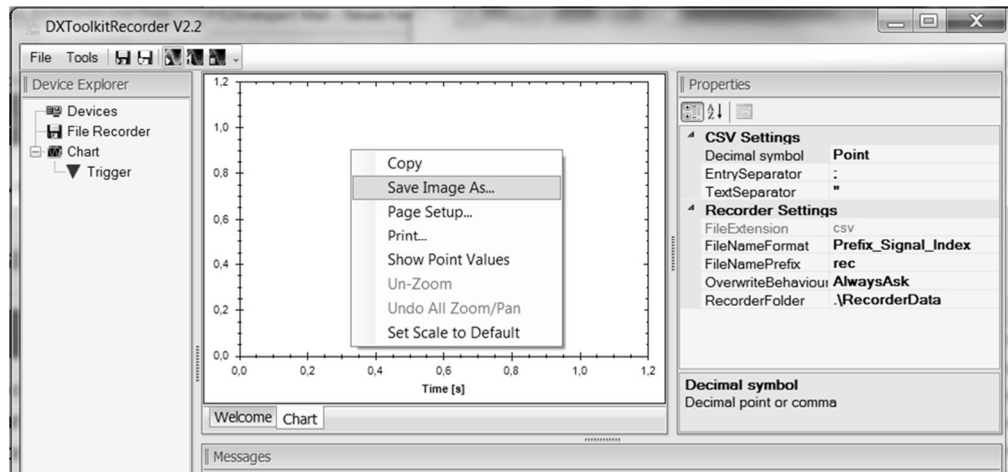
2.4 Supported TargetBoards / MotorController

The SPI Toolkit is able to work together with every controller that possesses a SPI unit. So it is necessary to include only the SPIRecorder.c module into the target controller application (please refer to chapter 6 SPIRecorder protocol). A Version for MLX81200 is to be found in the installation folder. Further versions for other controllers (like TI's C2000 series) are available by contacting the under info@drivexpert.de.

2.5 Data export and integration possibilities

The SPI Toolkit could be well integrated in your workflow in two different ways.

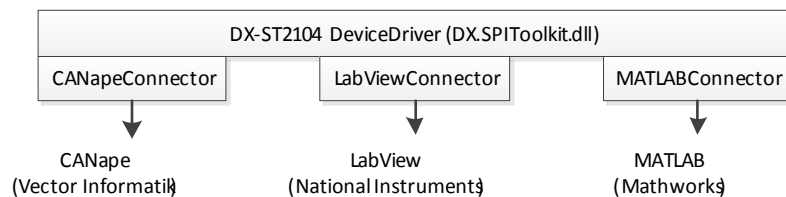
1. Data export from DXToolkitRecorder



Picture 2-5: Export feature via DXToolkitRecorder

Via chart context menu a direct connection to Office tools like Word could be used. The actual chart content could be exported into a few picture formats like .emf, .png, .jpg, .gif, .tif or .bmp. It is also possible to copy the picture direct from the software chart. With the help of the File Recorder data could be logged over a required time interval and saved as a .csv file for processing with Excel or MATLAB.

2. Integration of the DX-ST2104 DeviceDriver-DLL



Picture 2-6: Integration of the DX-ST2104 DeviceDriver DLL

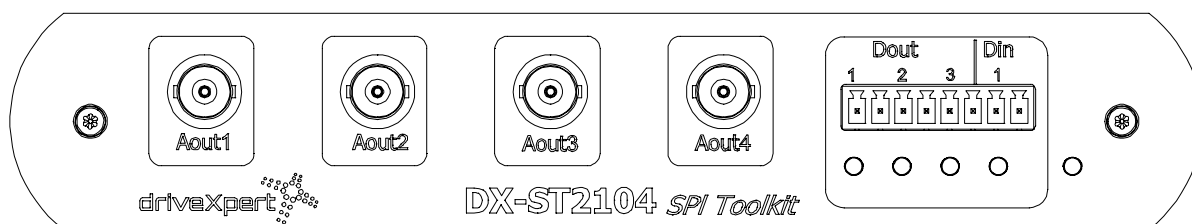
A direct integration in tools like CANape, LabView or MATLAB is possible via using DeviceDriver-DLL (DX.SPIToolkit.dll). This DLL is to be found in the DXToolkitRecorder installation directory. A detailed description could be ordered from info@drivexpert.de. For a complete integration another step, the Connector DLL, between the DeviceDriver DLL and the tool is necessary, but not embodied in standard scope of supply. Please contact us in case of interest.

3. Technical data

Table 3-1: Technical data of the DX-ST2104

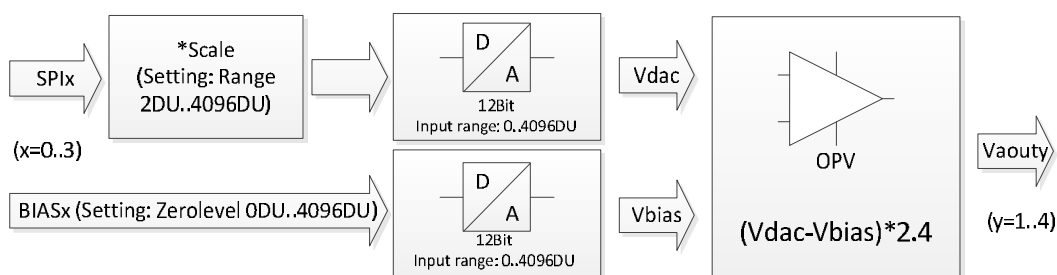
Parameter		MIN	TYP	MAX	Unit
I _{Base}	Basic current consumption		220		mA
+IP	current consumption peripheral		50		mA
+I _{com}	current consumption SPI communication		10		mA
=I _{Tot}	Total current consumption		260	270	mA
VCC1, VCC2	USB supply voltage 1 and 2; When using the power only USB (VCC2), VCC1 won't be loaded.	4.5	5	5.5	V

3.1 Connectors DX-ST2104 SPI Toolkit (Front)



Picture 3-1: Front connectors

3.1.1 Analog outputs Aout1..4



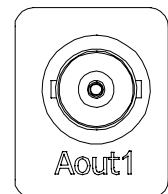
Picture 3-2: Aout channels (simplified block diagram)

Table 3-2: Technical data Aout1..4

Parameter		MIN	TYP	MAX	Unit
Vdac	Output voltage DAC data channel	0		4.096	V
Vbias	Output voltage DAC Bias channel	0		4.096	V
	DAC resolution (12Bit)		4096		DU
	DAC gain		1		mV/DU
	OPV-gain		2.4		
Vaout	OPV output voltage (at VBIAS = 2.048V=2048DU)	-4.92		4.92	V
RSeries	Aout serial output resistor		560		Ω

Table 3-3: Signal assignment Aout1..4

Connector	Designation	Meaning	Colour
1	Aout1	Analog output	Red
2	Aout2	Analog output	Yellow
3	Aout3	Analog output	Green
4	Aout4	Analog output	Blue



3.1.2 Digital inputs and outputs

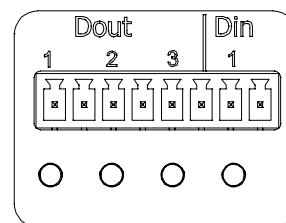
Table 3-4: Technical data Din, Dout

Parameter		MIN	TYP	MAX	Unit
Dout 1..3 (Open Collector output)					
Rpullup	Internal PullUp resistor		5.1		k Ω
Iout	Max. current Open Collector output			40	mA
RSeries	Serial output resistor		300		Ω
Din1 (digital input)					
VIH	High level input voltage		2.7		V
VIL	Low level input voltage		1.5		V
VI	Input voltage range	0		5.5	V
RTP	Resistor RC-lowpass		1		k Ω
CTP	Capacitance RC-lowpass		1		nF

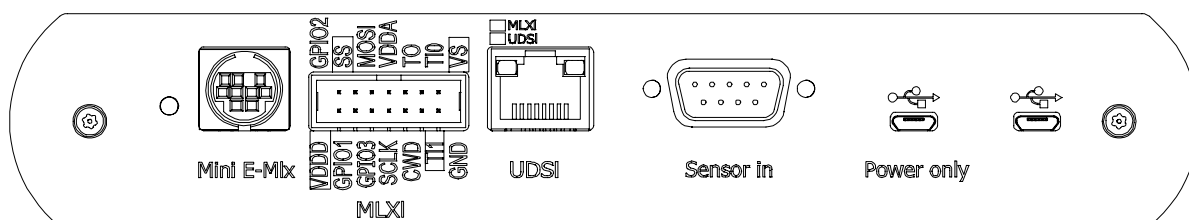
The use of digital inputs/outputs is indicated with the LEDs placed under the corresponding connector.

Table 3-5: Signal assignment Din, Dout

Connector	Designation	Meaning
1	Dout1	Digital output
2	Dout2	Digital output
3	Dout3	Digital output
1	Din1	Digital output



3.2 Connectors DX-ST2104 SPI Toolkit (back)



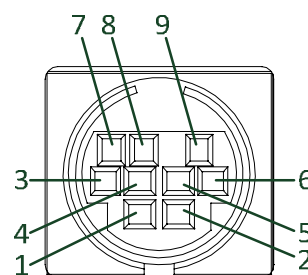
Picture 3-3: Back connectors

3.2.1 Mini E-Mlx connector

Connector for Melexis Mini Emulator.

Table 3-6: Signal assignment Mini E-Mlx connector

Pin	Designation	Meaning
1	GND	Ground
2	TIO	Digital MUST output (LSB)
3	Vout0	Programmable power supply 0
4	LINMUST	LIN or analog MUST output
5	TI1	Digitale MUST output (MSB)
6	Vbat	Car battery
7	Vout1	Programmable power supply 1
8	T0	Test output
9	Vout2	Programmable power supply 2



Mini E-Mlx

For detailed information please refer to the Melexis „Mini E-Mlx Hardware“ document.

3.2.2 MLXI and UDSI connector

Table 3-7: Technical data for MLXI and UDSI connector

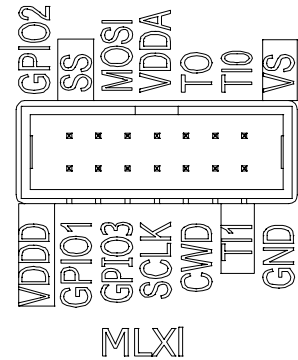
Parameter		MIN	TYP	MAX	Unit
FSmp	Sampling rate/transfer rate			20	kSmps/s
UDSI differential SPI input signals					
VIH	High level input voltage	2			V
VIL	Low level input voltage			0.8	V
VIC	Common mode input Voltage			±7	V
RTerm	terminating resistor		1		kΩ
MLXI single ended digital inputs					
VIH	High level input voltage		3		V
VIL	Low level input voltage		1.5		V
VI	Input voltage range	0		VCC+0.5	V
MLXI single ended outputs					
VO	Output voltage		3.3		V
RSeries	Serial output resistor		1		kΩ
MLXI analog inputs for voltage monitoring and Mini E-Mlx programming interface					
VDDA	Voltage level		5	12	V
VDDD	Voltage level		3.3	12	V
VS	Voltage level		12	24	V

3.2.2.1 MLXI connector

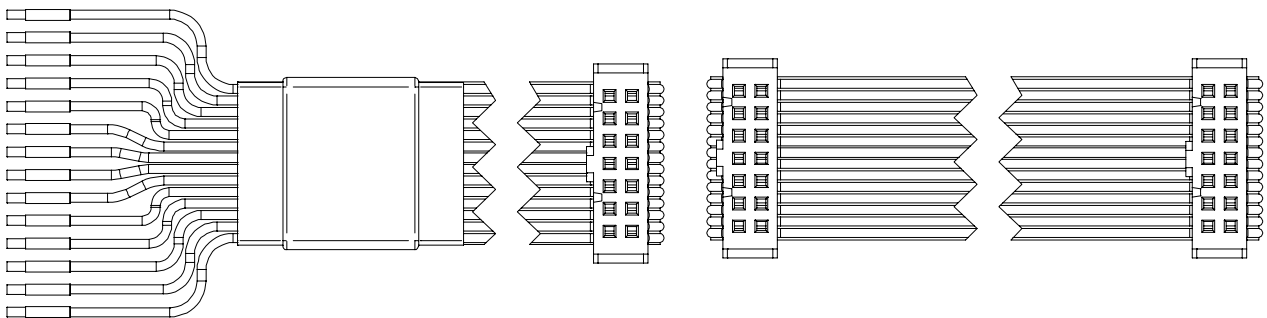
Melexis Interface connector SPI Master and SPI Toolkit or UDSIDriver circuit board.

Table 3-8: Signal assignment MLXI connector

Pin	Designation	Meaning	Colour
1	VS	Supply voltage	White
2	GND	Ground	Brown
3	TI0	Testpin Input 0	Green
4	TI1	Testpin Input 1	Yellow
5	TO	Testpin Output	Grey
6	CWD	WatchDog Disable	Pink
7	VDDA	Supply voltage (analog)	Blue
8	SCLK	Serial Clock (SPI)	Red
9	MOSI	Master Out Slave In (SPI)	Black
10	GPIO3	General Purpose Input /Output 3	Violet
11	SS	Slave Select (SPI)	White
12	GPIO1	General Purpose Input /Output 1	Brown
13	GPIO2	General Purpose Input /Output 2	Green
14	VDDD	Supply voltage (digital)	Yellow



The connector's coloured labels around the MLXI connector go together with the colours on the MLXI cable. The following picture shows both, the MLXI cable_b with an open end on the left and the MLXI cable_a with a closed end to the right.



Picture 3-4: Two types of MLXI cable

3.2.2.2 UDSI connector

UDSI connector for the UDSIDriver circuit board and the SPI Toolkit (numbering beginning from left).

Table 3-9: Signal assignment for the UDSI connector

Pin	Designation	Meaning
1	SCLK+	Serial Clock (SPI)
2	SCLK-	Serial Clock (SPI) inverted
3	MOSI+	Master Out Slave In (SPI)
4	MOSI-	Master Out Slave In (SPI) inverted
5	+5V	Positive voltage supply (UDSI)
6	GND	Ground
7	SS+	Slave Select (SPI)
8	SS-	Slave Select (SPI) inverted

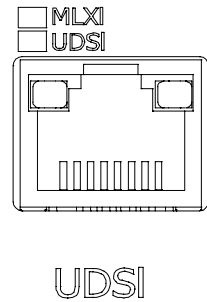


Table 3-10: Meaning of LEDs for UDSI connector

LED	Designation	Meaning of LED states	
		On	Off
Left	Active connector for communication with TargetBoard	UDSI connector	MLXI connector
Right	Not used		

3.2.3 Sensor in connector

SUB-D 9 connector position sensors and the SPI Toolkit.

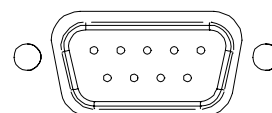
Only incremental encoders with differential output are supported. To use sensors with single ended outputs the SensorAdapter circuit board has to be used.

Table 3-11: Technical data for Sensor in connector

Parameter		MIN	TYP	MAX	Unit
Differential input signals					
VIH	High level input voltage	2			V
VIL	Low level input voltage			0.8	V
VIC	Common mode input Voltage			±7	V
Position sensor supply					
VSens	Sensor supply voltage		VCC		V
ISens	Sensor supply current			100	mA

Table 3-12: Signal assignment for Sensor in connector (pin numbering corresponds to standard assignment)

Pin	Designation	Meaning
1	A+	Track A positive (differential input)
2	B+	Track B positive (differential input)
3	I+	Index positive (differential input)
4	Home+	Stopp sensor (single ended input)
5	GND	Ground sensor supply
6	A-	Track A negative (differential input)
7	B-	Track B negative (differential input)
8	I-	Index negative (differential input)
9	+5V	5 V sensor supply (max. 100mA)



Sensor in

3.2.4 Micro USB connector (Power only)

USB connector for alternative voltage supply via USB

Table 3-13: Signal assignment for USB (Power only)

Pin	Designation	Meaning
1	VBUS	Supply voltage VCC2
2	UDM	- Not assigned
3	UDP	- Not assigned
4	GND	Ground
5	SHIELD	Ground



Power only

3.2.5 Micro USB connector

USB connector for data transmission between SPI Toolkit and PC

Table 3-14: Signal assignment for USB

Pin	Designation	Meaning
1	VBUS	Supply voltage VCC1
2	UDM	Data Minus
3	UDP	Data Plus
4	GND	Ground
5	SHIELD	Ground



When powering the SPI Toolkit via one of the USB connectors, this will be indicated by the LED on the front's right side.

3.3 UDSIDriver circuit board

1	Test pin array 1
2	Test pin array 2
3	GND pins
4	Switch
5	User Interface
UDSI	Connection to DX-ST2104; SPI-Data transfer signal assignment see 3.2.2.2 UDSI connector
MLXI	Connection to TargetBoard; Signal assignment see 3.2.2.1 MLXI connector
Mini E-Mlx	Connection to Mini E-Mlx for TargetBoard programming

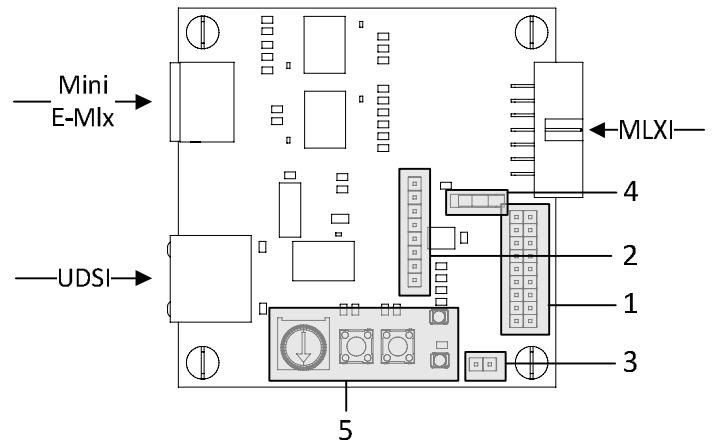


Table 3-15: UDSIDriver switch functions

Switch		
Switch	Enable	Disable
WATCHDOG	Analog Watchdog active	(Default) CWD-Pin is put to GND. Analog Watchdog inactive, necessary for Debugging

Table 3-16: Technical data for UDSIDriver

Parameter		MIN	TYP	MAX	Unit
I_UDSI	UDSI current consumption		40		mA
VCC	USB supply voltage	4.5	5	5.5	V
MLXI/UDSI connector (refer to Table 3-7)					

3.3.1 Functions

1. MLXI connector (galvanic insulated for PC safety)
2. Mini E-Mlx connector Melexis controller programming and debugging
3. UDSI transmitter: SPI signals are transmitted differentially to the DX-ST2104
4. UART to USB converter for a direct connection between TargetBoard and PC (also galvanic isolated)

The UDSI circuit board is an additional circuit for TargetBoards with Melexis controllers. It enables the use of the SPI Toolkit under conditions with electro magnetic distortion which is present in different testing scenarios. Like the DX-ST2104 it offers a MLXI connector for SPI signals and for programming via Melexis interface. In contrast to the SPI Toolkit all signals are galvanic isolated.

Another advantage is the differential transmission of SPI signals. This improves the disturbance behavior and allows longer cables between the test motor and the PC.

In contrast to the UDSIDriver circuit board Rev 2.1 combined with the DX-ST2104 Rev 2.2 the galvanic isolated side is not supplied by USB. Since UDSIDriver circuit board Rev. 2.2 and DX-ST2104 Rev 2.3 power is supplied via UDSI connector.

3.3.2 Using the UI

During development of embedded systems it is necessary to control the target. In case of there is no PC software for communication (LIN / CAN), the UDSI User Interface (LEDs, buttons and trimming poti) could be used to interact with the target board.

So it is possible to initiate motor start / stop etc. via using the UDSIDriver UI.

3.3.3 Checklist programming and debugging

The UDSI circuit board could additionally be integrated as a galvanic isolation into the Melexis toolchain consisting of the Mini E-Mlx and the Melexis Interactive Debugger.

In case of programming and debugging faults, please check the following points:

1. USB cable connected to the UDSIDrivercircuit board? TargetBoard voltage turned on? The yellow LED on UDSI-Connector must shine.
2. All test signals connected to the TargetBoard TI0, TI1, TO, CWD, GND and VDDA are needed.
3. All switches on UDSIDriver circuit board in default state?
4. Check Melexis Interactive Debugger configuration. Under Tools Target & Engine Mini E-Mlx and the correct Target have to be chosen.

If programming mode is active, the green UDSI connector LED is shining. When there was any system component crashing down, the Mini E-Mlx and the TargetBoard voltage have to be turned off. In this case the debugger should be closed and everything has to be reconnected and reactivated.

If there is still a programming fault, something could be defective. Maybe the TargetBoard's PullDown resistors are too low (please refer to chapter "Programming series electronics with 1k PullDown at the test signals").

3.3.4 Further configuration possibilities of the UDSIDriver circuit board

Schematics for UDSIDriver circuit board are listed in the appendix. These information are necessary to configure the board.

3.3.5 Reduce programming signal VDDA

The programming voltage differs between several Melexis controllers from 3.3 V up to 5 V. That's why it is advised to let the automatic voltage recognition via VDDA be enabled.

To reduce one signal, the programming voltage could be adjusted via the voltage divider R8-R9 (labeled as VDDIO in the schematic). Then the signal VDDA is not needed anymore. Please pay attention to the voltage T+5V on UDSIDriver circuit board to be 5.5 V.

3.4 SensorAdapter circuit board

1	Test pin row RM2,54 for position sensor connection
2	Terminalblock for position sensor connection
SensorOut	SubD-9 Female Connector for SPI Toolkit connection

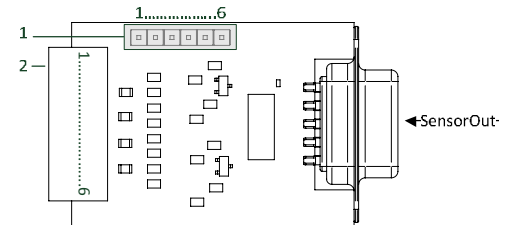


Table 3-17: SensorAdapter signal assignment, test pin row and terminal block

Pin	Designation	Meaning
1	+5V	Supply voltage (max. current 100mA)
2	A	Track A
3	B	Track B
4	I	Index
5	Home	Stopp sensor
6	GND	Ground

Table 3-18: Technical Data for SensorAdapter

Parameter	MIN	TYP	MAX	Unit
Single Ended input signals				
VIH	High-level input voltage	2		V
VIL	Low-level input voltage	0.8		V
VI	Input voltage range	0	5.5	V
RTP	Resistor RC-lowpass	1		kΩ
CTP	Cpacitance RC-Lowpass (not placed)	TBD		nF
Position sensor supply				
VSens	Supply voltage	VCC		V
ISens	Supply current		100	mA

3.4.1 General function

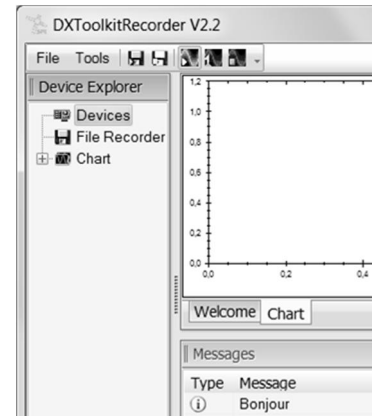
The SensorAdapter circuit board enables the connection of an incremental position sensor with single ended signals to the SPI Toolkit. This board converts from single ended to differential. Only incremental sensors with differential interface are able to be connected directly to the DX-ST2104.

It is possible to place individual RC combinations for lowpass filtering to improve signal quality. Schematics can be found in the appendix.

4. Software installation

4.1 DXToolkitRecorder installation

All software components for SPI Toolkit are installed from the package **DXToolkitRecorder_Setup_Vxx.exe** that is stored on the Toolkit's USB stick or could be downloaded from <http://www.drivexpert.de/index.php/download.html>.



4.2 SPI Toolkit software components

The following components belong to the installation package:

Table 4-1: SPI Toolkit software components

Software	Description
DXToolkitRecorder	Windows application to control the DX-ST2104. Enables data acquisition and visualization if SPI signals and a few measurement channels.
SPIRecorderModule	Programming examples in C code, that you have to include into your software project (refer to chapter "SPI Recorder Protocol"). In Mlx81200 BLDC_V3.0_Demokit this module has already been integrated.
FirmwareImage DX-ST2104	HexFile (*.S) with the DXToolkitRecorder firmware version.

4.3 System requirements

The DXToolkitRecorder is able to run on every Windows operating system with Microsoft .Net Framework 4.0.

Table 4-2: System requirements for .Net Framework 4.0

Client operating system	Minimal version and further software
Windows 7	Windows 7 RTM
Windows Vista	Windows Vista RTM
Windows XP	Windows XP Service Pack 3 (SP3) Internet Explorer 6.0 SP1 Windows Installer 3.1 or higher

An internet connection is necessary to install missing Windows components.

4.4 Software update

The latest SPI Toolkit software version can be downloaded from our softdist server from the following link:

<http://www.drivexpert.de/index.php/download.html>

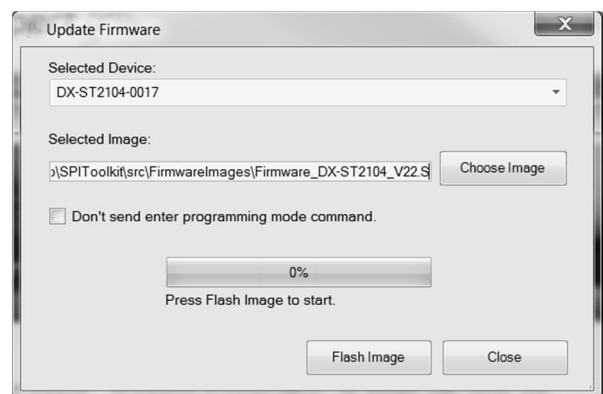
4.5 Flashing firmware

With a new version of DXToolkitRecorder the firmware normally has to be updated. This could be done as follows:

Menu->Tools->Update Firmware

4.5.1 Flash preparation

1. Please disconnect every cable except the DX-ST2104 USB cable before flashing
2. Selection of the device serial number
3. Selection of the image (*.S)
4. The button „Flash Image“ starts the flashing process



4.5.2 Flashing process

1. If the DX-ST2104 is in programming mode, the LED for DOUT1 is shining continuously
2. As soon as data transfer begins, LED DOUT1 and LED DOUT2 start shining continuously
3. After a successful flashing process the message „Flashing: 32864Bytes finished“ will appear. An error will be indicated via LED DOUT3 you will get the message „Flashing failed.“ The process has to be repeated after that.

4.5.3 Repeated flashing after error occurrence

After a flash abort because of transmission errors or an unrecognized device, it could be necessary to set the device into programming mode manually.

1. First of all the device has to be opened (only the screws in the back and on the Sensor in connector have to be unscrewed)
2. While the USB cable is still connected, please press the PB1 button to manually enter programming mode.

Now LED DOUT1 is shining continuously.

After that the device could be flashed in the same way mentioned above. **But now the checkbox „Don't send enter programming mode command.“ has to be checked in the DXToolkitRecorder software.**

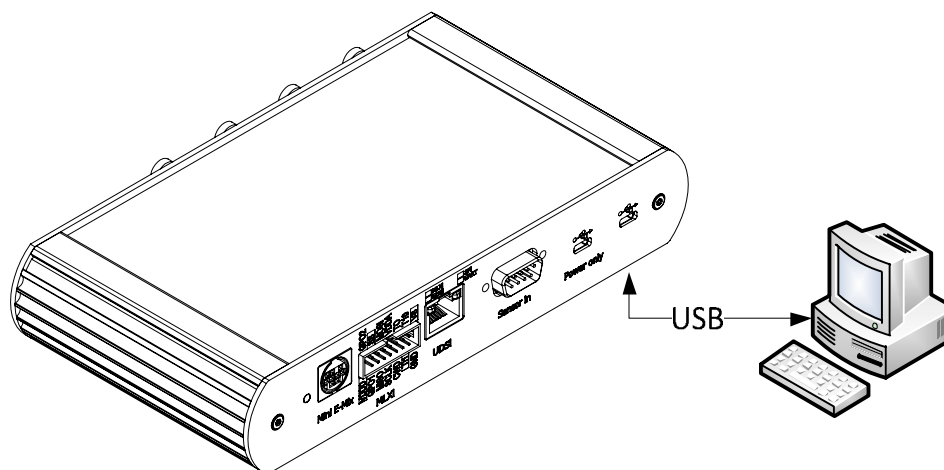
5. SPI connection establishment

The following chapter describes the basic work flow to physically connect the hardware and manually configure the software that is necessary to establish a SPI data connection from a TargetBoard to a PC for data acquisition and visualization.

5.1 Hardware

To receive data via SPI Toolkit two types of connection, SPI Toolkit---PC and SPI Master---SPI Toolkit can be realized.

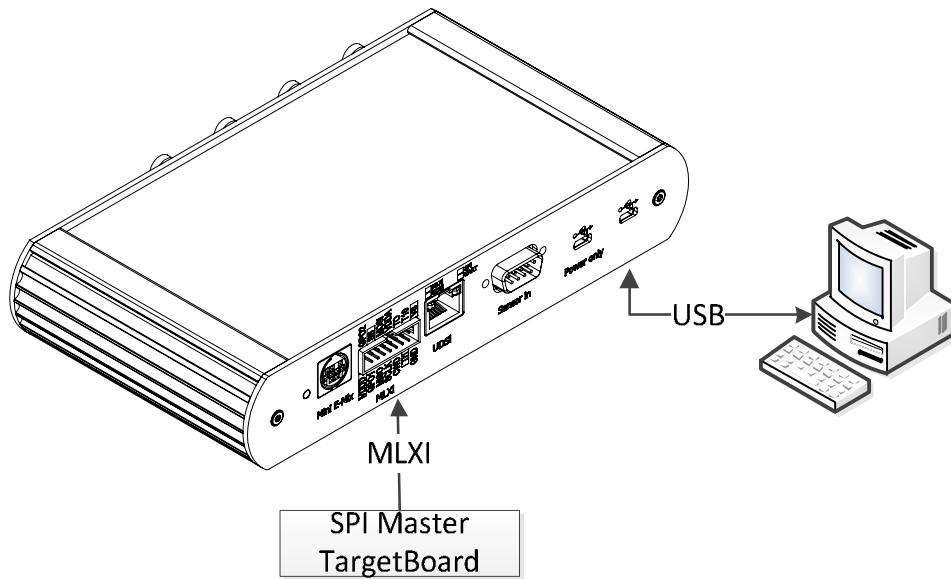
5.1.1 Connection SPI Toolkit---PC



Picture 5-1: USB connection between SPI Toolkit and PC

To establish a physical connection between SPI Toolkit and a PC, the SPI Toolkit has to be connected to the PC via the right Micro USB connector on the back side. Then the SPI Toolkit is displayed as a recognized device if driver installation has been successful.

5.1.2 Connection via MLXI connector (SPI Master---SPI Toolkit)



Picture 5-2: MLXI connection between SPI Master and SPI Toolkit

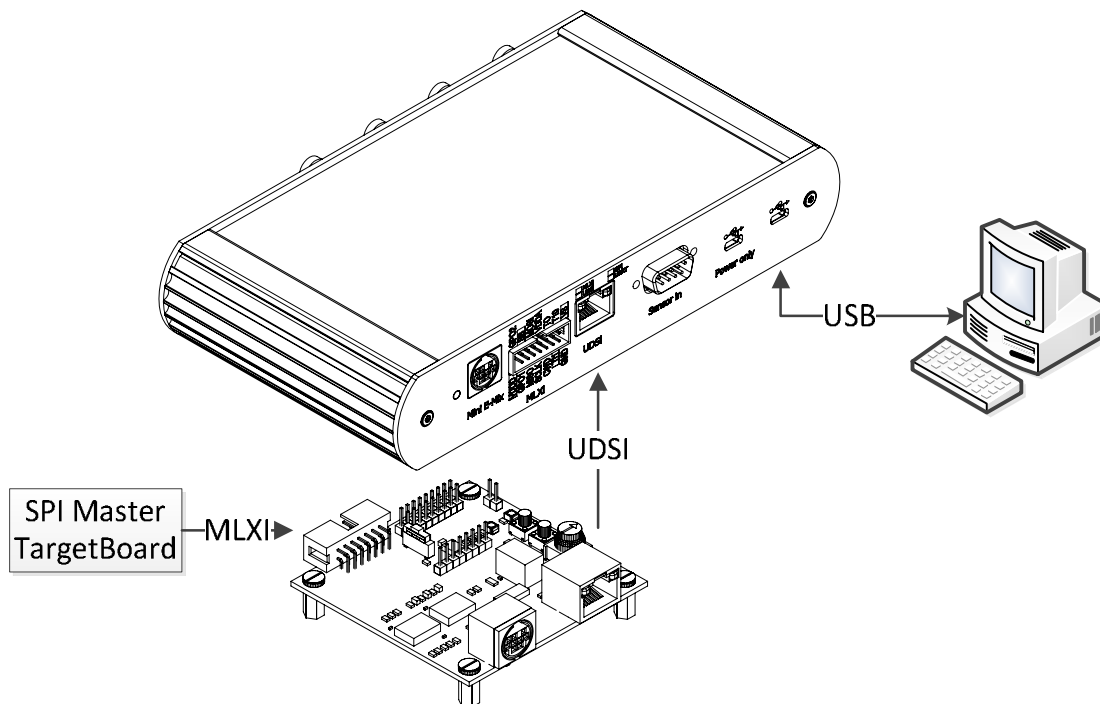
For a connection via MLXI connector the cable must be plugged to the TargetBoard that is operating as a SPI Master. The MLXI cable colours go together with the colouring scheme around the MLXI connector on the back side. Please choose the right cable (MLXI cable_a / MLXI cable_b). The following table will show the most important Pins for a correct SPI data transmission.

Table 5-1: SPI Pins

Pin	Designation	Colour
2	GND	Brown
8	SCLK	Red
9	MOSI	Black
11	SS	White

The MISO signal is not required, because data is only transferred from SPI Master to SPI Slave.

5.1.3 Connection via UDSI connector (SPI Master---SPI Toolkit)



Picture 5-3: UDSI connection between SPI Toolkit and UDSIDriver circuit board

The UDSI connector enables the connection of the UDSIDriver circuit board for decoupling SPI signals from SPI Toolkit. To do so the UDSIDriver circuit board has to be connected to the SPI Toolkit via the Ethernet flat ribbon cable. For UDSIDriver circuit board power supply use the USB power plug.

Now the MLXI cable is used to connect the UDSIDriver circuit board with the SPI Master with the correct colour assignment.

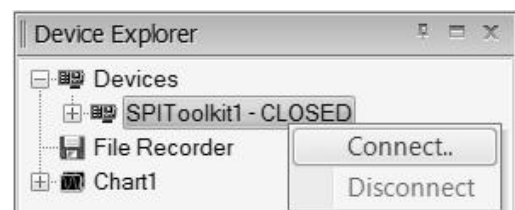
To use the UDSIDriver circuit board please select the „UDSI (Universal differential serial connector)“ from the dropdown list under the “SPI Connector” tab in your DXToolkitRecorder software. Usually the MLXI connector is selected as default configuration.

5.2 Software connection establishment with DXToolkitRecorder

After the physical connection has been established for SPI Toolkit---PC and SPI Master---SPI Toolkit, now the steps for organizing a software connection will be described below.

5.2.1 Connection to the plugged SPI Toolkit

For acquisition, visualization and recording SPI data please start DXToolkitRecorder.



If the plugged SPI Toolkit was recognized, it appears in the Device Explorer as a new node under the entry „Devices“. The connection state is indicated by one of the following four labels.

Table 5-2: SPI Toolkit states

Label	Meaning
UNPLUGGED	Will appear after the device was successfully connected once, but is not physically connected at the moment
CLOSED	Will appear after the device was successfully connected, but there was no channel opened by the DXToolkitRecorder software
OPEN	Will appear after the device was successfully connected and a channel was opened by the DXToolkitRecorder software
IN USE	The device was found, but the program was not able to open a channel

To establish a connection between SPI Toolkit and DXToolkitRecorder (open channel), the corresponding node under "Devices" has to be right clicked to open the context menu. Select "Connect" to open a channel. After that the SPI Toolkit state is indicated by OPEN.

To disconnect (close a channel), choose "Disconnect" from the context menu. Now the state changes into CLOSED. Data transmission will be terminated after that.

To configure one of the two connector variants and to setup the SPI properties use the DXToolkitRecorder's "SPI" tab. With the help of the dropdown list under „SPI Protocol Mode“ you can select a transmission mode. Per default the SPI Toolkit is able to communicate via standard SPI protocol.

If you want to refresh your new settings, press the button "Send Settings".

After the hard- and software connection is established, the SPI Master has to start sending SPI data.

Now the DXToolkitRecorder is able to receive and process incoming SPI data. To prove the correctness, select the corresponding SPI Toolkit node under „Devices“ with a left click and have a look to the field behind "DataRate" in the property window.

To visualize received data, you only have to drag the desired channel (node under your SPI Toolkit device) and to drop it into the graphical chart on the "Chart" tab with your left mouse button. Make sure, that the toolbar button "Start trigger (cont.)" was pressed for continuous data transfer and visualization.

5.3 Method for connection problems

In case of USB problems Windows gives a lot of hints, that can help troubleshooting. But the reactions differ from one system to each other.

Keep in mind that different controllers can have their own configuration possibilities with respect to the SPI signal timing. Also the length of cables, additional capacitances or different kinds of ground connection can highly influence SPI signal quality.

Please contact us in case of problems:

Martin Greiner

E-mail: mgr@drivexpert.de

Telephone: +49 3677 7996693

5.3.1 DX-ST2104 is not recognized under Windows or DXToolkitRecorder

5.3.1.1 USB driver not installed

In this case Windows will show the device as unknown.

Please repeat the installation with respect to all necessary USB drivers. The latest DXToolkitRecorder version and USB drivers can be downloaded from the softdist server under the following link:

<http://www.drivexpert.de/index.php/download.html>

5.3.1.2 USB port supplies not enough current

If there are too many devices connected the mainboard's supplying power can be not sufficient. The USB specification guarantees 250 mA for every working USB port. Windows will also prompt this. Another possible scenario is a system crash down. In this case follow the steps below:

- a) Disconnect all unneeded current consumer from the USB bus
- b) Disconnect position sensors from the SPI Toolkit
- c) Use another USB port
- d) Power the SPI Toolkit with the external USB power supply. This is done via using the SPI Toolkit's second USB connector (Power only)

5.3.1.3 DX-ST2104 firmware defect

After faulty flash processes Windows will give you an error message after connecting the DX-ST2104.

5.3.2 No reception or disturbed reception

If the DX-ST2104 is marked and opened in the "Device Explorer" you can observe the "DataRate" in the "Propertties" window. When there is no change in the „USBFramesReceived“ value, make sure, that no SPI data has been arrived to the SPI Toolkit.

In this case the following points have to be proved:

5.3.2.1 Correct SPI connector selected?

UDSI or MLXI connector can be selected in the „SPI“ tab. When your new settings were saved successfully the MLXI's yellow LED will indicate it.

LED on: UDSI Connector is selected.

LED off: MLXI connector is selected.

5.3.2.2 UDSIDriver voltage supply

When you use the UDSIDriver circuit board the yellow LED on the UDSI connector must shine. Otherwise the USB voltage supply is not connected or the used USB port is not active.

5.3.2.3 Vary „SPI Protocol Mode“

Four possible SPI modes to interpret SPI signals are available on the "SPI" tab. If „DataRate“ still doesn't change, please reset the mode to the first entry and check the other points.

5.3.2.4 Data output on TargetBoard side

The TargetBoard should output a ramp or a fixed value unequal to zero. All of the three SPI signals should be proved for correctness on an oscilloscope. SPI data rates greater than 1MHz require 1:10 probes. Furthermore check the ground cable.

5.3.3 „DataRate“ changes, but the chart is still empty

Do the following steps:

- a) Setup the chart trigger condition to „Automatic“ in the „Properties“ window and reset the zoom via the context menu (right click into the chart).
- b) Show SPI0 channel. This channel should always show data whenever the SPI Toolkit receives some of them. If only the MOSI signal is not connected a continuous stream of zeros will be received and shown on SPI0.

5.3.4 Data are overlapped with disturbances

If you use a MLXI connector, the UDSIDriver circuit board with galvanic isolation could also be used. After that check the SPI ground wire. Maybe the signal quality can be improved even without a ground. But in most cases it is better to use a ground wire as close as possible to the SPI outputs of the TargetBoard controller.

5.3.5 Sometimes negative time warps appear

This could be a hint for a buffer overflow at the DX-ST2104. Please check the TargetBoard data rate to be less or equal 20 kHz.

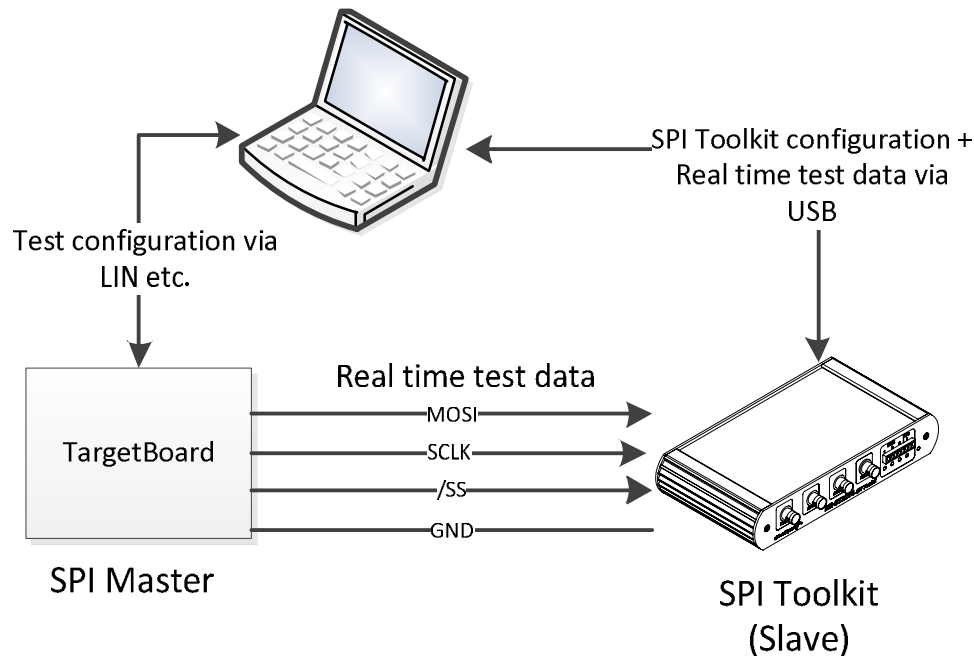
5.3.6 At block transmission mode offsets between SPI channels appear

Individual SPI-Frames are getting lost. The minimal distance between two SPI frames on the TargetBoard site has to be carried out.

5.3.7 Channel SPI0 is continuously showing zero

SPI signals SCLK and SS are operating well. But the MOSI is not connected or carries no data.

6. SPIRecorder protocol



Picture 6-1: Standard configuration

The SPIRecorder protocol is used to transmit real time test data from your TargetBoard to the DX-ST2104 via SPI interface. Data direction is only from Master to Slave. To configure data output another communication interface is necessary, because this is not done with the SPI Toolkit. So the DXToolkitRecorder is developed only for data acquisition as well. For TargetBoard configuration during a test a separate tool is necessary.

The following list shows all of the requirements that were most important for SPI Toolkit development:

1. Minimal computationally for the TargetBoard
2. Data output with 20 kSamples/s minimum, to send out data with PWM frequency
3. Implementation on different controllers with SPI unit
4. Possibility for real time transmission

From the efficiency point of view there are no features like CRC or parity bit for data safety and transmission error recognition. Data verification must be done by the user himself.

6.1 Protocol versions

The actual described SPIRecorder protocol version is 2.2 from April 2012. It is supported by DXToolkitRecorder 2.2.

New features compared with version 1.0:

1. Transmission of the data format in block transmission mode enables an automatic recognition and visualization via DXToolkitRecorder
2. Extended sample length from 12Bit to 15Bit
3. Header length reduced to 1Bit (toggle bit or synchronization bit)

6.2 Transmission modes and data formats

The SPIRecorder protocol supports two transmission modes.

1. Single Data Transmission (or real time transmission)
2. Block Data Transmission

Between two frames a minimal distance of about 33 μ s has to be carried out in both transmission modes to prevent buffer overflows.

The average transmission rate must not exceed 20 kSamples/s, otherwise this would generate an overflow in the SPI Toolkit's transmission buffer.

6.2.1 Single Data Transmission

This mode is used to send only one variable with a frequency up to 20 kHz. On the TargetBoard this variable is not buffered while it is sent to the controllers's SPI unit. Arrived on the SPI Toolkit data will get a time stamp, that's why time information is very accurate.

Data length is limited to 15 bit. Data value will be interpreted as unsigned (MSBit first).

Table 6-1: Data frame Single Data Transmission

Bit	Name	Description
15	ToggleBit	Changes its value after every frame. Data will be interpreted as single data. This bit is used for Single and Block Data Transmission distinction
14..0	DataBits	15Bit unsigned data

6.2.2 Block Data Transmission

In block mode whole arrays are transmitted. Every block starts with a 16 bit header frame, after that data frames follow. At every new header frame the ToggleBit is changed. The maximum block length depends on the block's data format. The minimum block length is about two SPI frames. Currently the SPI Toolkit supports five different data formats.

Table 6-2: Header frame Block Data Transmission

Bit	Name	Description
15	ToggleBit	If this bit changes after every block, data will be interpreted as block data
14..0	BlockFormatBits	Indicates the block's data format 0 = Multiple Samples Format unsigned 1 = Multiple Variables Format unsigned 2 = BlockBEMF Format 3 = Multiple Samples Format signed 4 = Multiple Variables Format signed

Table 6-3: Data frame Block Data Transmission

Bit	Name	Description
15	ToggleBit	If this bit changes after every block, data will be interpreted as block data
14..0	DataBits	15 bits with data

Table 6-4: Data formats of Block Data Transmission

Bit value	Name / enum Name	Description
0	Multiple Samples unsigned/ BDF_MULTISAMPLE_UINT15	Data bits are 15 bit unsigned values. Every data is shown on channel SPI0. Maximum block length: 16 bit
1	Multiple Variables unsigned/ BDF_MULTIVARIABLE_UINT15	Data bits are 15 bit unsigned values. Different values are shown on different SPIx channels. Maximum block length: 12 bit
2	BlockBEMF/ BDF_BLOCKBEMF	Special format for BEMF data transmission with flags: Bit14: not used Bit13: discarded Bit -> Channel SPI0 Bit12..10: motorstate Bits -> Channel SPI1 Bit9..0: BEMF Data -> Channel SPI2 Maximum block length: 16 bit
3	Multiple Samples signed/ BDF_MULTISAMPLE_INT15	Data bits are 15 bit signed values. Every data is shown on channel SPI0. Maximum block length: 16 bit
4	Multiple Variables signed/ BDF_MULTIVARIABLE_INT15	Data bits are 15 bit signed values. Different values are shown on different SPIx channels. Maximum block length: 12 bit

6.3 SPIRecorder module interface description

Several SPIRecorder protocol implementations are available (as C modules) for different controllers development environments. The following described interface enables you to integrate this protocol in your own projects. One example is also shown in the BLDC_V3.0_Demokit for Mlx81200 controllers, that can be downloaded on the link below:

Downloadlink: <http://softdist.melexis.com/>

Registration is necessary.

6.3.1 BlockDataFormats enumeration

Defines all available block data formats used with `spi_SendBlockData()`.

6.3.2 spi_Init function

Initializes the SPIRecorder module and configures IO pins and SPI hardware unit. It is only called at the very beginning of your program.

C:

```
void spi_Init(void);
```

6.3.3 spi_SendSingleData function

Initiates a SPI transmission in Single Data Mode immediately. It is used to observe only one variable within an interrupt. If it is unbuffered, the actual value is getting lost, when SPI transmission hasn't already been completed. The minimum distance between two calls of about 33 μ s has to be carried out.

C:

```
void spi_SendSingleData(uint16 data);
```

Parameter:

`uint16 data` - 15 Data bits of Single Data frames value. The HighBit will be overridden by the ToggleBit.

6.3.4 spi_TransmitNext function

This function transmits the next SPI frame in Block Data Mode. It must be called in the main loop regularly and it is carrying out the correct temporal distance by itself.

C:

```
void spi_TransmitNext(void);
```

6.3.5 spi_SendBlockData function

This function starts a Block Data transmission. The header frame is transmitted immediately. Every other frame is sent out after the `spi_TransmitNext()` call. If previous data haven't already been sent completely, the block is aborted.

C:

```
Boolean spi_SendBlockData(BlockDataFormats format, uint16*  
_dataBlock, uint16 _blockSize);
```

Parameter:

`BlockDataFormats format` - Format of the block

uint16* _dataBlock - Array with block data. Only the least 15 Bit are transmitted. The HighBit is overridden by the ToggleBit's value.

uint16 _blockSize - Length of the array to be transmitted.

Return:

True - Transmission has been started

False - Last transmission was not completed

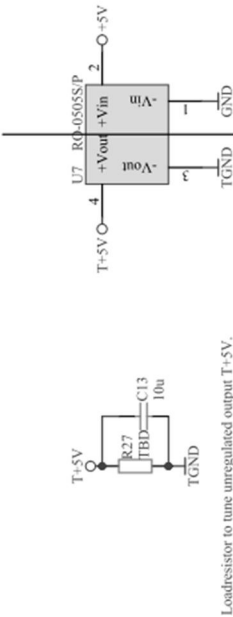
7. Appendix

Recommended Melexis interfaces are:

1. MLX81200
2. MLX81100
3. MLX81150
4. MLX81205/07/10/15

The following documents are about the UDSDriver circuit board and the SensorAdapter circuit board.

Nr.	Subject	Revision	Date
1	DX_SPIToolkit_UDSDriver_Schematic	2.2	November 2012
2	DX_SPIToolkit_UDSDriver_Assembly Drawing	2.2	November 2012
3	DX_ToolkitMA_SensorAdapter_Schematic	1.1	April 2012
4	DX_ToolkitMA_SensorAdapter_Assembly Drawing	1.1	April 2012

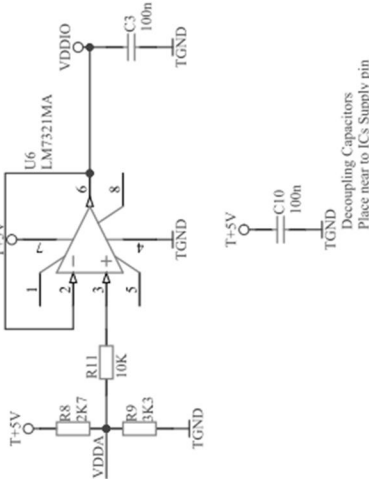


Loadresistor to tune unregulated output T+5V.

Melexis Cores: VDDA, CWD options

- Mulan I: no VDDA add pull up R8 to set VDDA on 3.3V, with CWD
- MulanII: VDDA 3.3V; no CWD
- MelexCM: VDDA 5V, with CWD
- Melex-FX: 3.3V; no CWD

Mulan has no VDDA output. Add optional R8 to set VDDIO to 3.3V

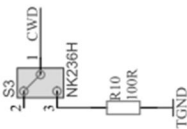
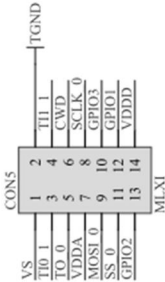


Decoupling Capacitors
Place near to ICs Supply pin

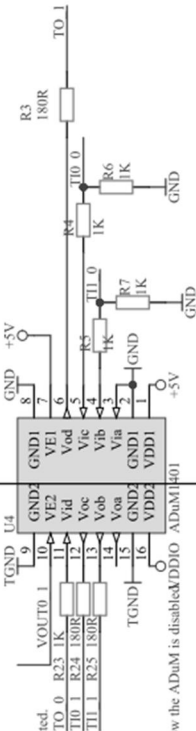
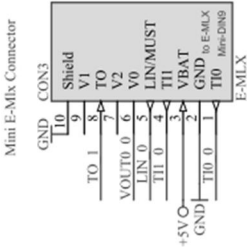
Melexis Cores: Featurelist

- Mulan I: Programming, SPI->Toolkit
- Mulan II: Programming, SPI->Toolkit
- MelexCM: Programming, SPI->Toolkit
- Melex-FX: Programming, SPI->Toolkit

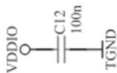
MLX1 Full (Testsignals, SPI, GPIO, Voltage monitoring)



Title <i>DX.UDS1Driver</i>		Revision:2.2	driveXpert
Size: A4	DX_SPTToolkit_UDS1Driver:SchDoc	Author: mgr / spu	
Print-Date: 23.11.2012	Modified (File Date): 19.11.2012	File:	



VDDIO is derived from VDDA. If VDDA is low the ADuM is disabled/VDDIO



Title <i>DX-UDSIDriver</i>		Revision: 2.2
Size: A4	Mix TestInterface.SchDoc	Author: mgr / spu
Print-Date: 23.11.2012	Modified (File Date): 23.11.2012	
File:		





Title <i>DX.UDS1Driver</i>		Revision: 2.2		
Size: A4	SPI.SchDoc	Author: mgr / spu		
Print-Date: 23.11.2012	Modified (File Date): 23.11.2012			
File:				

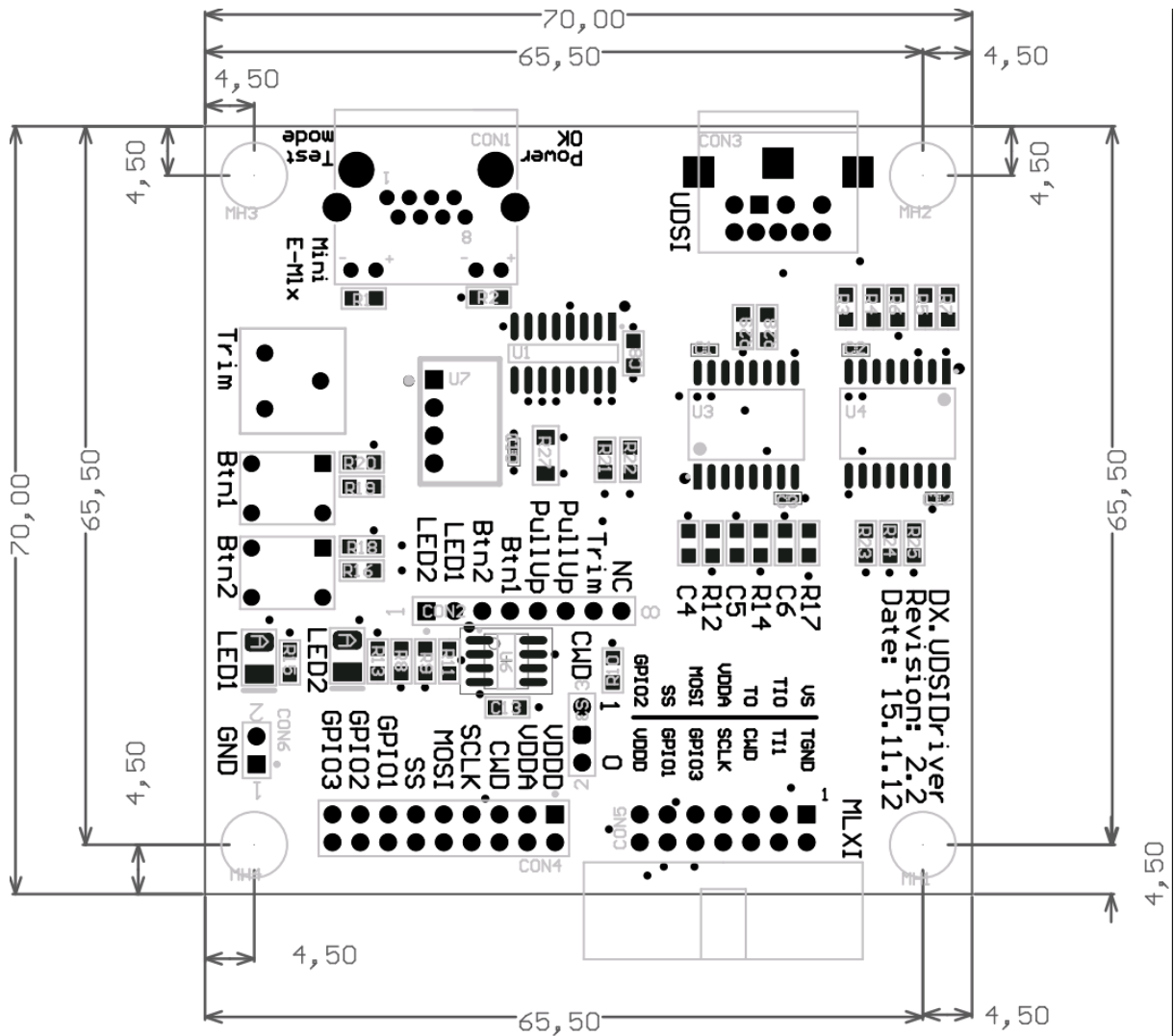
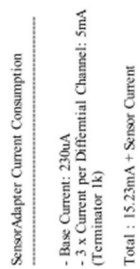


Abbildung 7-1 UDSDriver assembly drawing

42

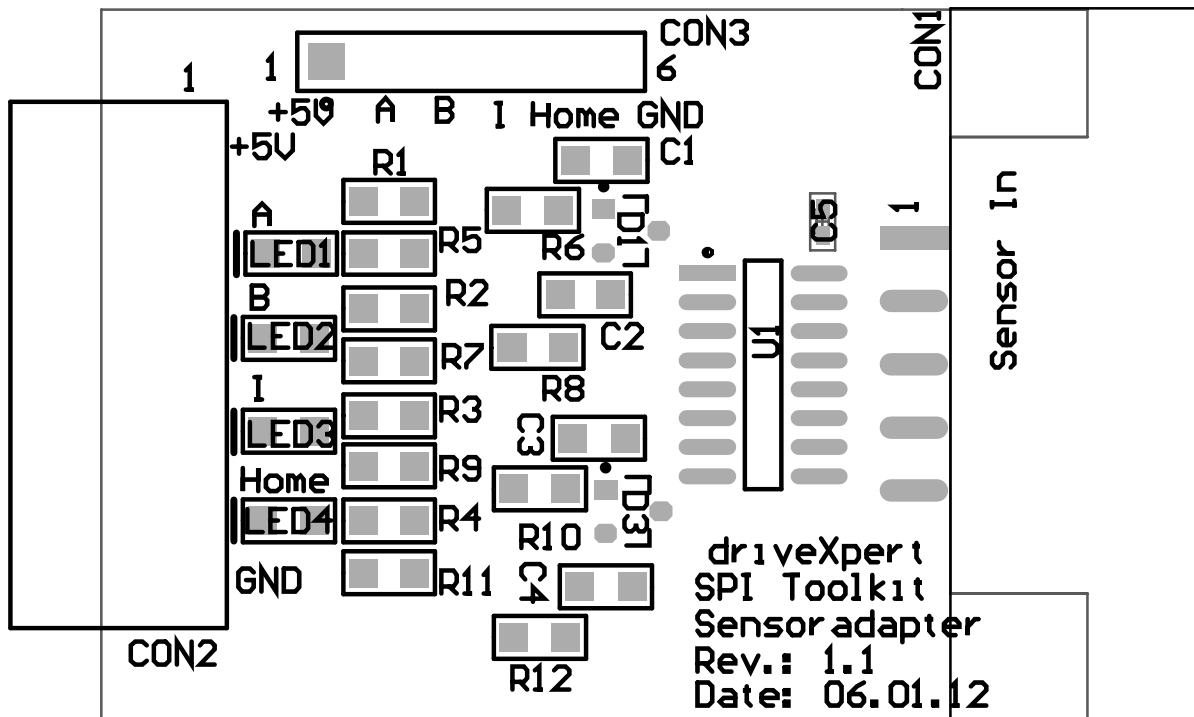


Abbildung 7-2 SensorAdapter assembly drawing



How to reach us:

Home Page: www.drivexpert.de

E-mail: info@drivexpert.de

Tel: +49 3677 7996690

Fax: +49 3677 7996691

Adresse:

driveXpert GmbH
Ehrenbergstraße 11
98693 Ilmenau, Germany