

Table of contents



1. Introduction

This document is intended to give a brief introduction of the different parts of the SW Development kit for the ASSP MLX81200. The aim of this document is to support a fast start up with this kit. Besides this document, several other important documents are necessary for a deeper understanding of more detailed development issues.

The most important documents related to the MLX81200 are:

MLX81200 Datasheet	-detailed description of the chip with all functions and features		
MLX82001_Product_Specification	-MelexCM Datasheet and underlying documentation of different blocks included in the MLX82001		
Melexis development system	-MLX Assembler, Linker, Obgen, Tabgen description		
GNU C-Compiler GCC	-GCC User's Manual, incl. AS, LD and Getting Started Doc		
Melexis emulator (flash programming SW)	-description of the Emulator		
Melexis emulator SW			
Mlx16 C-Debugger	-Debugger User's Manual		
MIx16 CPU Simulator	-description of the Simulator		
Mlx16X8 Data book	-explanation of the Mlx16X8 microcontroller core and the instruction set		
Melexis Lin Master documentation	-description of the Melexis USB LIN Master		
MelexCM LIN API (including required SW files)	-detailed description of the MelexCM LIN API		
MLX81200 BLDC Demo Kit firmware	-MLX81200_SWDesignDescription_x.pdf		
AppNote_MLX81100_MLX81200_Reflashing_on_module_x.pdf			
ApplNote_Flash_over_pin_LIN_x.pdf			



2. Contents of the MLX81200 SW Development kit

2.1. Hardware

Part	Picture
MLX81200	A HILLING TOTAL
Evaluation board	
Power board	



USB LIN Master	Melecis USB Lin Master version 10 Perconnected Perconnected Not mesage Vota Vo
Mini E-MLX Emulator	
BLDC Motor and hysteresis brake	
Hysteresis brake control board	

Tbl. 1 Contents Hardware



2.2. Software

Part	Function	Installation file	
Mlx16 Simulator/Emulator	Simulator/Emulator PC Software	Mlx16Setup_x_xx_x.exe	
Mlx81200x.mmf	Configuration file for Simulator/Emulator and Flash Programmer	Ish Mlx81200Conf_x_x_x.msi	
E-mlx MM/Mini E-MLX Emulator	Emulator Software and hardware driver	EmlxSetup_x_x.msi	
Mlx16 Interactive C Debugger	Melexis C Debugger	MDC_Setup_x_x_x.exe	
Flash Programmer Software for E-mlx MM and Mini E-MLX	Programming software for Flash memory	EMIxMMProgSetup_x_x_xx.msi	
Gnu Tool chain	Mlx16 C Compiler and associated tools	Mlx16_GCC_setup_x_xx_x.exe	
MLX81200 SW Platform	Includes general chip operation especially the LIN feature	Available on Softdist	
Melexis USB LIN Master Software	PC software for LIN Master	Comes with MLX USB LIN Master on CD	
MLX81200 BLDC Firmware	BLDC Firmware for the Demo kit setup	BLDC_DemoKit_Firmware_x.zip	
MLX81200 PC program	PC program to control the BLDC firmware via LIN	MLX81200_Demokit_PCProgram_x.exe	
MLX81200 SCT	PC program to configure, build und flash the BLDC-Demokit firmware	MLX81200_SCT_x.exe	
MLX81200 Software Source code	BLDC Firmware sources for the BLDC Demokit	BLDC_x_Demokit.zip	

Tbl. 2 Contents Software



3. How to use the MLX81200 SW Development kit

The following table introduces the philosophy of the MLX SW Development kit and gives an overview about the different options for development with the kit.

Task	Hardware	81200 specific hardware	Required software
Software development in Assembler			- Editor - Mlx16 Melexis Assembler
Software development in C			- Editor - Mlx16 GCC C Compiler
Software Simulation in Assembler	- PC		- Mlx16 Simulator
Software Simulation in C	- PC		- Mlx16 C Debugger
In-Circuit Emulation in Assembler	- PC - Mini E-MLX Emulator	- Evaluation board	- Mlx16 Simulator - Mlx16 Emulator - 81200x.mmf configuration file
In-Circuit Emulation in C	- PC - Mini E-MLX Emulator	- Evaluation board	- Mlx16 Simulator - Mlx16 Emulator - 81200x.mmf configuration file
Quick HW Test	- PC - Mini E-MLX Emulator	- Evaluation board	 MIx16 Simulator MIx16 Emulator 81200x.mmf configuration file Demo Software for the Evaluation board
Configure, build and program via pin LIN	-PC -Mini E-Mlx Emulator or -USB-LinMaster 2.0	-Evaluation board -MLX81200 IC with loader support	-Mlx16 GCC Compiler -MLX81200 SCT

Tbl. 3 How to use the SW Development kit



4. The Software Kit

4.1. Software Evaluation flow

The documentation for the development tools listed above is included with those tools and is outside the scope of this document. Despite that, a brief flow of the software development for the MLX81200 is given below in Fig. 1. Software can generally be written in Assembler or in C.

4.1.1. Software Development in Assembler language

Assembler source file(s) can be written with all common non-formatting editors. The code is then compiled with a Melexis supplied assembler, which is customized for the Mlx16X8 instruction set and supports all standard features of other commercial compilers.

The linker will merge the object files generated by the assembler and create a final file (*.hex, *.cod). This final *.hex file can be used in two different ways:

- for simulation purposes with the MIx16 Simulator
- for in-circuit debugging/testing (programmed to Flash)



Fig. 1 Flow of the Software Development using Assembler and Melexis tools



4.1.2. Software Development in C language

Gnu has also developed a C compiler, Assembler and Linker. This tool chain supports all known features of other common compilers. The creation of C source file(s) can be achieved with any common non-formatting editor. The code is then compiled and linked with the above mentioned Gnu tools (see Tbl. 2) With this methodology:

- All C language based features can be used
- Assembler parts (previously compiled to object file level) can be included
- ROM and In-circuit emulation is possible on C instruction level
- <u>NOTE:</u> The object files of Gnu and Melexis supplied Assembler are not compatible. Therefore the Gnu provided Assembler must be used, in case Assembler sources should be included with a Gnu C source.



Fig. 2 Flow of the Software Development using the Gnu C Compiler tools



4.2. The installation of the Software Kit

4.2.1. Installation of the tools

All tools run under WinXP®. Win9x, WinNT® and Win2000® are not supported. All programs use the standard windows installer of WinXP®.

The following tools have to be installed:

- EmlxSetup_x_x_x.msi
- MIx16Setup_x_xx_x.exe
- MDC_Setup_x_x_x.exe
- EMIxMMProgSetup_x_x_xx.msi
- Mlx81200Conf_x_x_x.msi
- MIx16_GCC_setup_x_xx_x.exe
- MLXLinMaster_Setup.exe
- MIxLinDebug.exe
- MLX81200_Demokit_PCProgram_x.exe
- MLX81200_SCT_x.exe

Emulator Software and hardware driver Mlx16 Simulator Software Mlx16 Interactive Debugger Flash Programmer Software Configuration file, consists of chip specific settings Mlx16 C Compiler USB LIN Master Software Melexis USB LIN Master PC program to control the BLDC firmware via LIN PC program to configure, build und flash the BLDC-Demokit firmware

x – Revision number

Demo Software can be copied at any top level directory:

•	Software_Platform_MelexCM_x_x_x.zip	Software platform including the Demo Software
•	BLDC_DemoKit_Firmware_x_x.zip	firmware for the Evaluation setup
•	BLDC_x_Demokit.zip	the source code of the BLDC DemoKit firmware

4.2.2. Directory structure of the Software Tools

It is recommended to use the "standard" installation option. After installing with this installation option, the following path settings and directory structure appears:

C:\Programs\Melexis\

stem Documentation
embler)
nd manual



The release of Mx16 GCC suite installs C:\mlx16-gcc\	in the following directory tree:
bin	contains executable files.
Lib	contains libgcc library and include files and compiler specs
mlx16	contains libmlx16 library and include files; start-up module; linker command files and memory map.
libexec	additional executable files.
config	configuration file and script for MIx16 Simulator.
docs	documentation.
examples	sample programs.
C:\Programs\Common Files\Melexis Sh	ared\Config\81200
81200.mmf	Configuration file for Simulator, Emulator and Flash Programmer
C:\Programme\LIN Commander\LinCom	nmander.exe
LinCommander.exe	a sample interface program that can be used to debug via the LIN bus when developing an application.

4.2.3. Directory structure of the Demo Software

The setup works with relative paths, so it can be copied into any top level directory.

NOTE: Please make sure that directory path does not contain any spaces

It contains the following folders:

\bin	Utilities	
\config	Makefile configurations	
\doc	Documentation related to general MelexCM platform software	
\include	Common libraries include directory	
\lib	Common libraries (.a), library object files and linker files	
\mmf	Test controller files and tools	
\libsrc	Root of the common libraries sources	
\libsrc\lib	Source code of the common libraries	
\libsrc\LIN	LIN library source code	
\libsrc\math	Math library source code	
\projects	Root of the projects sources	
\projects\Examples	Projects which are independent from analogue chip periphery	
\projects\81200	Root of the MLX81200 project	
\projects\81200\include	The MLX81200 specific declarations (analogue port names etc.)	
\projects\81200\src	00\src The MLX81200 common versions of the product specific functions	
	(analog_trimming, power_down and vectors.S) and projects	
\projects\LINLoader	LIN pin loader source code (LIN mode or standalone)	

Tbl. 4 Folder description



5. Hardware Kit

5.1. Configuration

- Evaluation board in conjunction with the Mini E-MLX Emulator
- Used for In-Circuit debugging
- Flash programming possible
- LIN Communication possible



Fig. 3Arrangement of Evaluation board in conjunction with Power board, Mini E-MLX Emulator , BLDC motor and Melexis USB LIN Master



5.2. General

The purpose of the HW kit is the development of software for the MLX81200 BLDC Motor Controller. Using this HW kit, the device can be evaluated in a detailed fashion.

The evaluation system is composed of an evaluation board and the power board. The idea of the evaluation system is to have two stand-alone boards:

- A standard board (evaluation board) is used for all possible applications. It consists of a socket for the chip, several pin headers for all signals from the chip and interface connectors to the Emulator, LIN-Bus and the application board (power board).
- A customized application board (power board) has to be designed specifically for a certain application by the user. It is possible to connect this power board on 32-pin connector. The sample power board consists of 3 half Bridges with N-FET-transistors, 2 shunts for possible current supervision and interface connectors to Vbat and the BLDC-Motor.

5.3. Evaluation board

5.3.1. Evaluation board - jumper settings



Fig. 4 Location of Connectors on Evaluation board



JMP1	short open	: connects Vbat to the chip. : disconnects Vbat from the chip
JMP2	short open	: connects CWD-pin to GND : CWD-pin of the chip is active. NOTE: the external capacitor for analog Watchdog is active.
JMP3	short open	: disables the external regulator NPN transistor for higher VCC loads. : enables the external regulator NPN transistor for higher VCC loads
JMP5	1-2 2-3	: enable low pass filter R55/C15 and voltage divider R5/R50 for pin T : enable only the voltage divider R5/R50 for pin T
JMP6	1-2 2-3	: enable low pass filter R66/C16 and voltage divider R6/R60 for pin SW5 : enable only the voltage divider R6/R60 for pin SW5
JMP7	1-2 2-3	: enable low pass filter R77/C17 and voltage divider R7/R70 for pin SW6 : enable only the voltage divider R7/R70 for pin SW6
JMP8	1-2 2-3	: enable low pass filter R88/C18 and voltage divider R8/R80 for pin SW7 : enable only the voltage divider R8/R80 for pin SW7
S2 con	sists of t	he jumper JMP9 and JMP10 [.]
JMP9	short open	 : the connection between the chip and the emulator is split : the connection between the chip and the emulator is established, the chip can be accessed by the emulator, if the emulator is not plugged the CPU is executing the flash program
JMP10	short open	: CPU does not execute the flash program : CPU is executing the flash program, if no emulator is connected and JMP9 is shorted
JMP11 +JMP1	short 2	: the pull up resistors R21 and R22 are connected to the test interface inputs (necessary if the E-mlx MM programmer is used)
	open	: the pull up resistors are disconnected (default for the Mini E-Mix programmer)
JMP21	short open	: supply the additional voltage regulators inside the Mini E-Mlx programmer (required) :disconnect VBAT from the test interface (required for the E-mlx MM programmer) NOTE: If the E-Mlx MM programmer is connected and the JMP21 is shorted the diode D4 can be destroyed.

5.3.2. Functionality of the Evaluation Board

Main Power Switch S1 selects between supply connector of the evaluation board or the supply connector of the power board.

IMPORTANT: If the evaluation board is supplied via the power board, the power supply connector of the Evaluation board should **NOT** be connected! Otherwise the board or the power supply will be damaged!

The PCB only requires a DC voltage of +12...+18V which is applied via the supply connector of the evaluation board. This is valid if the evaluation board is used stand-alone without the power board. In case the board is used alongside the power board, the power board delivers the supply voltage for both PCBs.

Both the evaluation board and the power board are protected against reverse polarity.



5.3.3. Evaluation board – Phase signal filters



Fig. 5 Position of the filter components on the EVB

The resistors R13, R14, R15 and the diodes D13, D14, D15 protect the IC against negative current and voltage from the motor coils. The resistors are increase the discharging time of the high side n-channel MOSFET transistor gates.

The phase signals on the T, SW5, SW6 and SW7 pins can be reduced with the voltage dividers R5/R50, R6/R60, R7/R70 and R8/R80.

With the low pass filters R55/C15, R66/C16, R77/C17 and R88/C18 fast disturbances can be filtered out. The default value for the resistor is 150 ohm and for the capacitor is 10nF.



5.3.4. Evaluation board - Jumper ring

The jumper ring enables the user to either connect or disconnect all signal lines to the chip. By removing jumpers, the circuit from the chip to the application hardware can be disconnected e.g. for measuring current or applying external signals. All jumpers are described with the specific pin names on the board.

5.3.5. Connection between Power Board Port and Evaluation Board

Connector	32	160	4
Connector	52		
on Evaluation			
board			
Pin	Namo	Description	Direction
1	GND	Ground	Direction
2	GND	Ground	
3	GND	Ground	
4	W	IC-Pin 24	Input
5	HS2	IC-Pin 25	Output
6	SW0	IC-Pin 23	Input/Output
7	GND	Ground	
8	SW1	IC-Pin 22	Input/Output
9	V	IC-Pin 31	
10	HS1	IC-Pin 30	Output
11	GND	Ground	
12	SW2	IC-Pin 21	Input/Output
13	U	IC-Pin 33	Input
14	HS0	IC-Pin 34	Output
15	GND	Ground	•
16A	VBAT S2	IC-Pin 40	Input
16C	VBAT_S1	IC-Pin 39	Input
17	V5EXT	IC-Pin 2	Output
18	Т	IC-Pin 37	Input
19	GND	Ground	
20	LS2	IC-Pin 27	Output
21	LS1	IC-Pin 28	Output
22	LS0	IC-Pin 36	Output
23	GND	Ground	
24	EN_VS	Bootstrap output from IC-Pin 15 -CLKO	Output
25	SW3	IC-Pin 20	Input/Output
26	GND	Ground	
27	GND	Ground	
28	GND_S2	IC-Pin 42	Input
29	GND_S1	IC-Pin 41	Input
30	GND	Ground	
31	VBAT	unprotected voltage from Power board	Input
32	VBAT	unprotected voltage from Power board	Input

Tbl. 5 Connector ST1(male) on Evaluation board



5.3.6. Evaluation board – extension port ST3

Connector ST3 on Evaluation board	2 20 1 19		
Pin	Name	Description	
1	V5V or V5EXT	5V power supply or protected switchable V5ext	
2	VS	Protected voltage	
3	SW3	STB	
4	SW4	Wake-up interrupt signal (INH)	
5	GND	Ground	
6	IO5	Interrupt from CAN module	
7	GND	Ground	
8	IO4	SPI clock	
9	GND	Ground	
10	IO3	SPI MOSI	
11	GND	Ground	
12	IO2	SPI MISO	
13	GND	Ground	
14	IO1	SPI chip select	
15	GND	Ground	
16	Not used		
17	Not used		
18	Not used		
19	Not used		
20	Not used		

Tbl. 6 Connector ST3 on Evaluation board (for example a CAN extension board can be connected)

5.3.7. Evaluation board - Signal pin headers

Connector ST5 on Evaluation board	1 6 GND		
Pin	Name	Description	
1	IO0	IC Pin 1	
2	IO1	IC Pin 7	
3	IO2	IC Pin 10	
4	IO3	IC Pin 12	
5	IO4	IC Pin 9	
6	105	IC Pin 3	

Tbl. 7 Connector ST5 on Evaluation board



Connector ST7 on Evaluation board		1 9 GND
Pin	Name	Description
1	LS0	IC Pin 36
2	CP0	IC Pin 25
3	HS0	IC Pin 34
4	HS1	IC Pin 30
5	CP1	IC Pin 29
6	LS1	IC Pin 28
7	LS2	IC Pin 27
8	CP2	IC Pin 26
9	HS2	IC Pin 25

Tbl. 8 Connector ST7 on Evaluation board

Connector ST9 on Evaluation board	1 8 GND		
Pin	Name	Description	
1	SW0	IC Pin 23	
2	SW1	IC Pin 22	
3	SW2	IC Pin 21	
4	SW3	IC Pin 20	
5	SW4	IC Pin 19	
6	SW5	IC Pin 18	
7	SW6	IC Pin 17	
8	SW7	IC Pin 16	

Tbl. 9 Connector ST9 on Evaluation board

5.3.8. Evaluation board - Reset

Resets are always active low and can be achieved by the following options:

• Turning off the power switch on the Evaluation board resets all components of the Evaluation board as soon as the charge of the block capacitors falls below the reset threshold level

In case of working with the Mini E-MIx Emulator, there are two more reset options (this is valid only for the digital part of the chip, MelexCM):

- Sending a new software (*.x16 or *.elf file) to the Emulator will reset the MelexCM chip automatically
- Reset can also be achieved by the 'Reset' button of the PC Simulator/Debugger software.

IMPORTANT: Reset of the whole chip including the analog part can ONLY be achieved by turning off the power switch.



6. Quick start up

6.1. Using the C flow

Assembling/Linking:

- change to the .\libsrc directory
- open a command prompt and type: 'gmake clean' 'gmake install'
- change to the directory where the source code is situated, e.g. .\projects\81200\BLDC\
- open a command prompt and type: 'gmake clean' 'gmake all'

-> executable *.elf file is created in the same directory

-> executable *.hex file is created in the same directory

Simulation:

- Start the MLX Interactive Debugger '...\Melexis\MDC\MLXDBGW.exe'
- Select target 'MIx81200' and 'Simulator' from 'Tools' -> 'Target & Engine' menus as shown in Fig. 6
- Load the appropriate *.elf file 'File' -> 'Open'
- Run the *.elf files by pressing the 'Go' or 'Step' icon

Emulation:

- Connect the Mini EMIx Emulator to the PC via an USB cable
- Insert a MLX81200 chip to the appropriate socket on the Evaluation board
- Power up the Evaluation board and connect the Mini EMIx Emulator to the EVB
- Start the MLX Interactive Debugger '...\Melexis\MDC\MLXDBGW.exe'
- Select target 'MIx81200' and 'Mini EMIx' from 'Tools' 'Target&Engine' menus as shown in Fig. 6
- Load the appropriate *.elf file 'File' -> 'Open'
- Run the *.elf files by pressing the 'Go' or 'Step' icon

Choose	Target & Engine	×
Target:	Mix81200	×
Using:	Simulator	
	Simulator EMIx MM Mini EMIx	

Fig. 6 Interactive Debugger: Selecting Simulator or Emulator mode



	\$ En 🕘	(1)	ተብብ	1) 66 💭 🕅	, 8							
ile Map 🛛 📮 🕹	Disassemb	ly main.c nev	v_functions.c					4 Þ	×	Memory	dump	Ļ
File name 🔺	310	2D6E		444 	IO Interupt	Flag =(1	.<<5);		-	Address	0xf876	
in.n	311	and the second second			break;		0.699.699.0200			0000	λ 5 5λ	7
nain c	312			}						00002	5A A5	7
oports MelexCM,h	313			break;						0004:	00 00	1717
oports_custom.h	314		}							0006:	00 00	È la p
iew_functions.c	315	2074	ml Relea	seBuffer():						0008:	FA 21	·
n.h	316	2078	return:							000A:	FI 21	
n.c	317	32101	LCOULTY							DODC.	EC 21	
ashupload.c	318	2								0010:	DA 76	, v
n_tables.h	310									0012:	00 10	1
ni_Lininit.c	320	(* No	in Loon -							0014:	FF FF	
lu AutoAddPulse c	321	int w	ain()							0016:	FF FF	4.4
u DataTransmitted c	222		ain()							0010	09 00	
	344	х.								001C:	01 00	1
	323									001E:	00 00	L
eakpoints 🛛 🕂 🗙	324		unsigned	i int long AD	CValue1=0;					0020:	1D 2B	3 . +
	325									0022:	39 47	9G
ddre Add Id cor	326		/* LIN]	nitializatio:	n */					0024:	55 63	UC
ak pointe' liet	327		ml_Statu	us iStatus ;		11	return	value for Lin-Funkti	LC	0028	09 00	, q .
	328	2D7A	ml_LinIr	nit();						002A	11 15	
ddress Londi	329	23	//Set Li	n-IDs						002C:	19 1D	1
🗹 @ 0x2D7A	330	2D7E	iStatus=	ml_SetLinId(LIN_Status,	Ox01);	/*	LIN_Status -> initi	Ĺε	002E:	21 25	1%
~	331	2088	iStatus=	ml_SetLinId(LIN_ADCRead,	Ox02);	/*	ADC_Read -> initial	L:	0030:	02 26	, δε
	332	2092	iStatus=	ml SetLinId(LIN ADCConfig,	OxO3);	1 *	ADC Config -> init;	Lŧ	0032:	U3 B3	1.12
	333	2D9C	iStatus=	ml SetLinId(PWM Config,	Ox04);	1 *	PWM Config -> init;	L٤	0034:	07 73	
	334	2DA6	iStatus=	ml SetLinId(LED Switch.	0x05 1:	1 *	LED ON-Off -> initi	Lŧ	0038	02 B3	
	335	100000	//akt.iva	te Bus						003A:	16 47	G
	336	2080	iStatus	ml Connect ()						003C:	06 73	. s
	337	00001	inouran							003E:	06 4C	: .L
to Co Remove emove A	338		11							0040:	85 39	
	220	appal	afatoDia						•	0042.	07 40	с. Ц. с. Ц.
м 💷 🔯 🖾 🗌		4						,	•	0044.	07 40	2 M 1
Turning the second seco				Name	Value		T. X				1	
Context : main() line 32.	2		i i	IN ame	value			A FC00 X F801	PC	2D7A		
	1.200			PWM AKTIV	0					-		
Name Va	lue	Type	at at	LinFrameBuffer[1]	0 '\v00'		un gi	Y 1BDC S E16A	М	1723		
ADLValue1 365	/1162	unsigned long	Te te	In Status	0 100		un er			'		
207	4	int		LinFrameBuffer[5]	0 '\x00'		un vi	D 00000000				
istatus 207	4	Int		<< Click here to add	a new item >>		- 100	(22)				
				4			F		-		1	

Fig. 7 Melexis Interactive Debugger session



6.2. Programming a HEX file to the FLASH

- Connect the Mini E-MLX Emulator to the PC via an USB cable
- Insert a MLX81200 chip into the socket and power up the EVB
- Connect the Mini E-MLX Emulator (programmer) to the EVB
- Start the programmer software '...\Programmer\EMIxMMProg.exe'
- Select File\Open... to open the MIx81200 mmf file installed by MIx81200Conf_x_x_x.msi

E-Mix MM Programmer	_ _
File Help	
\Lambda No File loaded	Diagnostic
	A Hardware and software not yet checked
Open MMF File	<u>? ×</u>
Look in: 🗁 Mlx81200	🔽 🔇 🌶 📂 🖽 -
includ ib WMX81 WMX81	
Program Files Common Files Melexis Shared Config Milex1200	On LPT1 💌
File name States My Network Places	<u>Open</u>
Files of type: Must Mice Files (*.mmf)	Cancel
	Abort: Execute
Messages	

Fig. 8 E-Mlx MM Programmer: Path to the Mlx81200.mmf file

Select 'Tools\Options\Programming' tick checkbox 'Keep Supply between patterns'



Fig. 9 E-Mlx MM Programmer: Option "keep supply between pattern"



Mix81200	Memory Pattern Sequence Programmer	
	EMix MM	a 🗾
	Mini EMIx	
	Abort Diagnose	
essages		

• Select tab 'Programmer', choose from pull-down menu 'Mini E-mlx' on 'Autodetect'

Fig. 10 E-MIx MM Programmer: Chose the Mini EMIx

• Push the '*Diagnose*' button

-> This message should appear: >>> Info : Hardware successfully checked!

Mix81200	Memory Pattern Sequer	Memory Pattern Sequence Programmer					
	Mini EMIx	Mini EMIx • On AutoDetect •					
	Item	Description MiniEMIx.sys(1.7.0.0) ad 1, ver 3.0, high speed E-Mix VID_3E38/PID_A508 C.\Program Files\Common Files\Melexis Share 2s200fg256 2007/11/12 15:20:09 r 8A07001A 200mA					
Messages	Abort	Diagnose					
<pre>>> Echo : All EndPoim >> Echo : Toggle 8051 >> Echo : Load firmwa: >> Echo : Toggle 8051 >> Echo : Toggle 8051 >> Echo : All EndPoim >> Echo : Xilinx Rese >> Echo : Xilinx load</pre>	cs connected! Reset (01) re in Cypress Reset (00) cs connected! t sucessfull ed sucessfully cessfully checked!						
>>> Info : Hardware su							

Fig. 11 E-MIx MM Programmer: checking connected emulator



- Select tab 'Pattern' and go to pattern 'POWER_ON'
- Push the 'Execute' button -> in addition to D3 "Power on" D20 "E-MLX on" should light up

MIN81200 POWER POWER_ON POWER_OFF	Memory Pattern	Memory Pattern Sequence Programmer					
	Argument	Base	Value				
	Disassemble]	Stop				
Messages >>> Echo : Executing PO >>> Info : Action succe	WER_ON ssfully completed	8					

Fig. 12 E-Mlx MM Programmer: enable programmer interface

- Select 'Flash' from the tab 'Memories' sections
- Select the 'Program using HEX file' radio button
- Load a *.hex file and push the 'Execute' button
- Verify the programmed file by pushing the 'Compare Dump to HEX file' radio button and press 'Execute' again

💼 E-Mix MM Programmer - C:\Pro	gram Files\Common I	iles\Melexis Shared\Config\MLX81200 💶 🗖 🗙			
File Tools Help					
🖻 🌉 Mix81200	Memory Pattern Sequence Programmer				
Ram	Parameter	Value			
Flash		MEMORY			
NvRam	J TE Luddh	1024 bits			
lo lo		1024 bits			
Analogio		Address 0 (0x0), e.g Byte address 0x0			
	Size	256 (UXTUU) words of TU24 bits (32.0 KBytes)			
	V 🖸 Format	Little Endian			
	C Dump C	Frase C Program using Edited Data			
	i bang i i				
	Program using HEX file: C Compare Dump to HEX file:				
	C Dump and Compare to HEX file: 1				
	D:\projects\MLX81200\Software\GLC\BLDC_V1.4.1_DemoKit_				
	Lise Frase All				
	1 000 21000 74				
	Sh	Execute			
Managan Like way Manual					
Messages Memory View					
Reading HEX file					
Loading Memory	-				
>>> Warning : Using ERASEAL	L Data not in	HEX file will be erased			
>>> Into : UR					
The Description and	lated OF				
>>> Info - Ob	Teced or				
>>> Info ; Dump completed 0	K				
Reading HEX file	20				
Comparing file to memory					
>>> Info : No error found					
r		Elapsed: 21,506s 🔗 CRC locked			

Fig. 13 E-MIx MM Programmer: programming the flash



- Before disconnecting the Mini EMLX select tab 'Pattern' -> 'POWER_OFF'
- Push the '*Execute*' button
- D20 "E-MLX on" should now be turned of

E-Mlx MM Programmer - C:\Pro File Tools Help Mix81200	gram Files\Common Files 	Melexis Shared\Conl Jence Programmer	fig\MLX81200 💶 🔲 🗙
POWER POWER_ON POWER_OFF POWER_OFF	Argument Ba	se Value	Evecute
Messages >>> Echo : Executing POWER >>> Info : Action successfu	OFF lly completed		
 Pattern arguments		Elapsed: 0.020s	CRC locked

Fig. 14 E-MIx MM Programmer: disable programmer interface

- The evaluation board can now be switched off
- Mini E-MLX Emulator can now be safely removed

For generating *.hex code please refer to 6.1

7. MLX81200 Software configuration tool (SCT)

7.1. General

The Software configuration tool can be used to configure, compile and flash the BLDC-Demokit firmware version 2.0.

The Software configuration tool is not suitable to write a complete application. If the file structure is changed or the source files are modified, the tool will not work correctly.

Note: The save function will overwrite the files motorctrl.h, sysdef.h, ramppar.c, .\config\chip.mk and the application Makefile with a defined content. That means any modification will be lost.

The Software configuration tool does NOT include any source code for the MLX81200. The required firmware source files package BLDC_V2.0_DemoKit can be found on the Melexis Softdist server. The firmware needs to be unpacked to a local folder on the PC. Please make sure that directory path does not contain any spaces.

The SCT is divided into three major parts.

- Part 1: locate the demokit firmware sources
- Part 2: configure the motor control parameters
- Part 3: compile the firmware and flash the firmware into the IC via the pin LIN

A working loader inside the IC is necessary for a successful flash process. Melexis will deliver preprogrammed devices, which include an initial working loader code.

If the preprogrammed loader code was destroyed, the first step must be always the programming of the loader software via test interface. This step can only be executed via the programmer software or debugger using E-mlx Emulator.



For programming a HEX file to the flash via the test interface please refer to chapter 6.2

For further information please see:

- the description for the MLX81200 BLDC Demo Kit firmware
- Application Note how to program on module
- Application Note how to flash over pin LIN

7.2. Locate the demokit firmware sources and load the configuration

LX81200 SCT GUI1	
1elexis	1 11/121
oject name & path 2.Project Settings 3.Copy & compile project	
Project path D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta Browse	
Project name BLDC BLDC 1	
Target name 2	
4 read parameter 5	
Log Window]
Log Window]
Log Window check path found .//config//Chip.mk found .//config//Config.mk found .//config//Config.mk]
Log Window check path found .//config//Chip.mk found .//config//Config.mk found .//libsrc//Bakefile found .//libsrc//Bakefile found .//libsrc//Bakefile found .//libsrc//Bakefile found .//libsrc//Bakefile]
Log Window check path found .//config//Chip.mk found .//config//Rules.mk found .//ibsrc//Makefile found .//projects//LINLoader//Makefile found .//projects//B1200//include//sysdef.h	I
Log Window check path found .//config//Chip.mk found .//config//Dules.mk found .//config/Pules.mk found .//projects//Bl200//include//sysdef.h found .//projects//Bl200//include//motorctrl.h User has chosen :	I
Log Window check path found .//config//Chip.mk found .//config//Rules.mk found .//config//Rules.mk found .//projects//LINLoader//Makefile found .//projects//LINLoader//Makefile found .//projects//B1200//include//sysdef.h found .//projects//B1200//include//sysdef.h User has chosen : user has chosen project D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta\projec	I
Log Window check path found .//config//Chip.mk found .//config//Cules.mk found .//ibsrc//Makefile found .//projects//LINLoader//Makefile found .//projects//B1200//include//sysdef.h found .//projects//B1200//include//motorctrl.h User has chosen : user has chosen project D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta\projec]
Log Window check path found .//config//Chip.mk found .//config//Config.mk found .//inbsrc//Makefile found .//projects//Bl200//include//sysdef.h found .//projects//Bl200//include//sysdef.h found .//projects//Bl200//include//motorctrl.h User has chosen : user has chosen project D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta\projec	I
Log Window check path found .//config//Chip.mk found .//config//Eules.mk found .//inbsrc//Makefile found .//projects//B1200//include//sysdef.h found .//projects//B1200//include//motorctrl.h User has chosen : user has chosen project D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta\projec	I
Log Window check path found .//config//Chip.mk found .//config//Dules.mk found .//inbrc//Makefile found .//projects//Bl200//include//sysdef.h found .//projects//Bl200//include//sysdef.h found .//projects//Bl200//include//motorctrl.h User has chosen : user has chosen project D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta\projec	
Log Window check path found .//config//Chip.mk found .//config//Rules.mk found .//config/Rules.mk found .//projects//LINLoader//Makefile 6 found .//projects//B1200//include//sysdef.h found .//projects//B1200//include//motorctrl.h User has chosen : user has chosen project D:\projects\MLX81200\Software\Borland_CPP\SCT\BLDC_V2.0_DemoKit_beta\projec	

Fig. 15 MLX81200 SCT chose the path and load the configuration

- 1 Button "Browse" the path to BLDC firmware source 2
 - Project selector the current project name must be selected
- 3 Project name shows the current project name
- 4 Edit field "Target name" the name of the firmware
- Button "read parameter" the configuration will be read from the chosen directory 5
- Log Window each operation is shown in this window 6



7.3. Configure the demokit firmware

7.3.1. Test pulse configuration

MLX81200 SCT GUI1	
Melexis	Small things make a big difference.
Project name & path 2.Project Settings 3.Copy & compile project Motor Operation PWM_DUTY_CYCLE_MIN 20 * PWM_DUTY_CYCLE_MAX 253 * 7 MOTOR_LEAD_ANGLE_DIVISOR 14 * 9 OVERCURRENT_COUNTER_MAX 20 * 9 OVERCURRENT_THRESHOLD_ADC 83 * 9 LIN PWM No Interface 9 1	System settings IUMBER_OF_POLE_PAIRS 5 \pm 8 SLDC_MODE TRIPOLAR T BLDC_SENSOR SENSOR_LESS T SWM_PERIOD_FREQUENCY 20KH2 PID_CONTROLLER ENABLE T START_UP_BEHAVIOR TEST_PULSES WATCHDOG (digital) ENABLE T Speed regulator settings MOTOR_ROT_SPEED_MIN_RPM 300 \pm MOTOR_ROT_SPEED_MAX_RPM 4000 \pm 10 TARGET_MOTOR_ROT_SPEED_AFTER_START_UP_RPM 1400 \pm
Test pulse settings SSET Test pulse settings SSET	Motor startup Motor startup 12 MIN_PWM_DUTY_CYCLE_SU 42 MAX_PWM_DUTY_CYCLE_SU 192 INIT_VALUE_PWM_DC 96 STEP_SIZE_PWM_DC_SU 8 PWM_UP_DATE_COUNTER_MAX 3 MIN_MOTOR_STATE_TIME_SU 1060 MAX_MOTOR_STATE_TIME_SU 4770 INIT_VALUE_MOTOR_STATE_TIME 1 STEP_SIZE_MOTOR_STATE_TIME 1 MIN_TUP_DATE_COUNTER_MAX 1
Fig. 16 MI X81200 SCT – modify the pa	1.1.19.8 arameter of the BLDC-demo kit

- 7 Parameter set "motor operation"
- 8 Parameter set "System setting"9 Parameter set "Communication Interface settings"
- 10 Parameter set "Speed regulator settings"
- 11 Parameter set "Test pulse settings"
- 12 Parameter set "Motor start up test pulses"



7.3.2. Ramp configuration

lelexis	0	4	7	3 <		mai	l thi	ngs	mak	e	24	801		10.0 1	-
lectronic Integrated Systeme		C C C C C		Por las		i Dig	um	eren	ce.		- 53	2-01			
ject name & path 2.Project Settin	ngs 3.C	Сору & со	mpile pro	ject											
or Operation	-		e.	ala	Sys	stem settin	ngs					_			
M_DUTY_CYCLE_MIN	20		_		NU	MBER_OF	F_POLE_	PAIRS	5		_	8	-		
M_DUTY_CYCLE_MAX	253		1		BLC	C_MODE	ł		TRIPOL	AR _	BLDC	_SENSO	R SI	ENSOR_	LESS
FOR_LEAD_ANGLE_DIVISOR	14				PW	M_PERIO	DD_FREC	QUENCY	20kHz	1	PID_	CONTRO	LLER EI	NABLE	
RCURRENT_COUNTER_MAX	20				ST/	ART_UP_	BEHAVI	DR	RAMP	-	WAT	CHDOG (digital) El	NABLE	
RCURRENT_THRESHOLD	83														
DOUDDENT TUDEOUDLD ADD	r 1257	-			Sp	eed regula	ator settin	ngs							
RCORRENT_THRESHOLD_ADU	- 1921	-			¥P	cod rogaio									
munication Interface settings	-1		<u> </u>		MC)TOR RO	T SPEE	D MIN I	RPM			300	_÷		
munication Interface settings	• [9			TOR_RO	T_SPEE	D_MIN_I	RPM RPM			300 400		10	
MURRENT_THRESHOLD_ADI munication Interface settings N PWM No Interface			9			TOR_RO	T_SPEE	D_MIN_I	RPM RPM			300 400		10	
MCURHENT_INHESHOLD_AUT		n [9600	9 Baud	<u>ज</u>	мс мс ТА)TOR_RO)TOR_RO RGET_MI)T_SPEE)T_SPEE OTOR_F	D_MIN_I D_MAX_ OT_SPE	RPM RPM ED_AFTI	ER_STAF	RT_UP_F	300 400 PM 140	0 0 0 1 1	10	
HEURHENT_I HRESHULD_AU munication Interface settings V PWM No Interface	J_SPEEI	D 9600	9 Baud	¥	MC MC TA)TOR_RO)TOR_RO RGET_MI)T_SPEE)T_SPEE OTOR_F	:D_MIN_I :D_MAX_ ROT_SPE	RPM RPM ED_AFTI	ER_STAF	RT_UP_F	300 400 PM 140		10	
MCURMENT_INHESHOLD_AU munication Interface settings	۹_SPEEI	D 9600	9 Baud	Y)TOR_RO)TOR_RO RGET_MI)T_SPEE)T_SPEE OTOR_F	:D_MIN_I :D_MAX_ ROT_SPE	RPM RPM ED_AFTI	ER_STAF	₹T_UP_F	300 400 PM 140		10	
HOURHENT_INHESHOLD_AU munication Interface settings V PWM No Interface I Standard UN 2 x Y LIN Number of ramp steps	V_SPEEI	D [9600	9 Baud	Y)TOR_RO)TOR_RO RGET_MI)T_SPEE)T_SPEE OTOR_F	:D_MIN_I :D_MAX_ ROT_SPE	RPM RPM ED_AFTI	ER_STAF	ΥŢ_UP_F	300 400 PM 140		10	
Number of ramp steps	V_SPEEI	D 9600	9 Baud	¥)TOR_RO)TOR_RO RGET_MI	DT_SPEE DT_SPEE OTOR_F	:D_MIN_I :D_MAX_ ROT_SPE	RPM RPM ED_AFTI	ER_STAF	IT_UP_F	300 400 PM 140		10	
Number of ramp steps	v_SPEEI	D 9600	9 Baud 3	¥.)TOR_RO)TOR_RO RGET_MI	DT_SPEE DT_SPEE OTOR_F	:D_MIN_I :D_MAX_ ROT_SPE	RPM RPM ED_AFTI	ER_STAF	RT_UP_F	300 400 РМ 140		10	
HCURMENT_INRESHULD_AUT munication Interface settings Image: Standard Interface settings I Standard Interface settings <th>V_SPEEI</th> <th>D 9600 ■ 1</th> <th>9 Baud</th> <th>×</th> <th></th> <th>)TOR_RO)TOR_RO RGET_MI</th> <th>DT_SPEE DT_SPEE OTOR_F</th> <th>:D_MIN_I :D_MAX_ ROT_SPE</th> <th>RPM RPM ED_AFTI</th> <th>ER_STAF</th> <th>RT_UP_F</th> <th>300 400 PM 140</th> <th></th> <th>10</th> <th></th>	V_SPEEI	D 9600 ■ 1	9 Baud	×)TOR_RO)TOR_RO RGET_MI	DT_SPEE DT_SPEE OTOR_F	:D_MIN_I :D_MAX_ ROT_SPE	RPM RPM ED_AFTI	ER_STAF	RT_UP_F	300 400 PM 140		10	
HCURHENT_INRESHULD_AUT munication Interface settings Image: PWM No Interface I Standard UN 2x	v_SPEEI	D [9600 → 1 → 1	9 Baud 3	¥ [3]		ITOR_RO ITOR_RO RGET_MI	DT_SPEE DT_SPEE OTOR_F	D_MIN_I D_MAX_ ROT_SPE	RPM ED_AFTI	ER_STAF	11_UP_F	300 [400 PM [140		10	[14]
Number of ramp steps NUMBER_OF_RAMP_STEPS NUMBER_OF_RAMP_STEPS Ramp parameter array Index[014] pwmStartUp	19 [19	D 9600	9 Baud 3	[3] 112	[4] [4] 112	108_80 108_80 108_80 8681_41	DT_SPEE DT_SPEE OTOR_F 0TOR_F 161 222	D_MIN_I D_MAX_ ROT_SPE	RPM RPM ED_AFTI [8] 112	ER_STAF	RT_UP_F	300 400 PM 140 [11] 112	(112)	10 [13] 112	[14] 112
Number of ramp steps NUMBER_OF_RAMP_STEPS Ramp parameter array Index[014] pwmStartUp stepMotorStateTime (ms)	19 10 3180	D 9600 T 112 2915	9 Baud 3 [12] 112 2650	I3 112 2650	[4] 112 2650	ITOR_RO ITOR_RO RGET_MI	DT_SPEE DT_SPEE OTOR_F 0TOR_F 222 2650	D_MIN_I D_MAX_ ROT_SPE	RPM RD_AFTI ED_AFTI 112 2650	ER_STAF	IT_UP_F	300 400 PM 140 [11] 112 2650	(12) 112 2650	10 [13] 112 2650	[14 , 112 2650
HCURHENT_INRESHULD_AUT munication Interface settings PWM No Interface I Standard IN 2x I Mumber of ramp steps IIN NUMBER_OF_RAMP_STEPS INMBER_OF_RAMP_STEPS Ramp parameter array Index[014] pwmStartUp stepMotorStateTime (ms) Index[1529] Index[1529]	<pre>_SPEEI [19] [0] 3180 [15]</pre>	D 9600 → 1 112 2915 [16]	9 Baud 3 [112 2650 [17]	▼ I3 112 2650 I18]	[4] 112 2650 [19]	ITOR_RO ITOR_RO RGET_MI I I I I I I I I I I I I I I I I I I	DT_SPEE DT_SPEE OTOR_F 0TOR_F 222 2650 [21]	D_MIN_I D_MAX_ ROT_SPE	RPM RPM ED_AFTI 112 2650 [23]	ER_STAF	IT_UP_F	300 400 PM 140 140 140 140 140 140 140 140 140 140	(12) 112 2650 [27]	10 [13] 112 2650 [28]	[14] 112 2650 [29]
RCURMENT_INRESHOLD_AUT munication Interface settings PWM No Interface I Standard IN 2.x II Number of ramp steps IIN NUMBER_OF_RAMP_STEPS Ramp parameter array Index[014] pwmStartUp stepMotorStateTime (ms) Index[1529] pwmStartUp Index[1021]	19 19 19 3180 112	■ 9600 ■ 100 ■ 112 2915 ■ 112	9 Baud 3 [12] 112 2650. [17] 112	▼ I 3] 112 2650 [18] 112	[4] 112 2650. [19]	ITOR_RO ITOR_RO RGET_MI 33 2650 1112	0T_SPEE 0T_SPEE 0T0R_F 0T0R_F 222 2650 [21] 112	ID_MIN_I ID_MAX_ ROT_SPE	RPM RPM ED_AFTI 112 2650 [23] 112	ER_STAF	I112 I112 I112 I112 I112 I112	300 400 PM 140 140 140 140 140 140 140 140 140 140	(112) 112 2650. 112	10 [13] 112 2650 [28] 112	[14] 112 2650 [29] 112
RCURMENT_INRESHULD_AUT munication Interface settings I Standard IN 2x I Standard Index[0.14] pwmStartUp stepMotorStateTime (ms) Index[1529] pwmStartUp	19 19 19 3180 [15] 112 2550	D 9600 → 1 112 2915 (16) 112 2059	9 Baud 3 [2] 112 2650 [17] 112	 [3] 112 2650 [18] 112 	[4] 112 2650 [19] 112	ITOR_RO ITOR_RO RGET_MI (5) 33 2650 [20] 112 2050	Image: Control of the second	D_MIN_I D_MAX_ 30T_SPE 112 2650 [22] 112 2050	RРМ RРМ ED_AFTI [8] 112 2650 [23] 112 2650	ER_STAF	IT_UP_F	300 400 PM 140 [11] 112 2650 [26] 112 2055	(112) 112 112 112 112 112	10 [13] 112 2650 [28] 112	[14] 112 2650 [29] 112

Fig. 17 MLX81200 SCT - modify the parameter of the BLDC-demo kit

- 7 Parameter set "motor operation"
- 8 Parameter set "System setting"
- 9 Parameter set "Communication Interface settings"
- 10 Parameter set "Speed regulator settings"
- 13 Parameter set "Motor start up Ramp"



7.4. Save, compile and program the firmware

7.4.1. Configure the hardware in use

MLX81200 SCT				
File Window Configuration Help				
dose programmer an use Mini E-MLX use LINMaster 2.0 Microelectionic Integrated Systeme	er programmling	Small things make a big difference.	7180	16
1. Project name & path 2. Project Settin	ngs 3.Copy & compile project			
Step 3.1 - Save actual configuration Save Configuration Step 3.2 - Build new firmware	ution 15	14		
Build firmware	I Remove objects, Hex- an I I I Compile MelexCM libraries I Compile Loader	d ELF-files		
Step 3.2 - Upload firmware Load Firmware into the IC Compile process window				

Fig. 18 MX81200 SCT – configure the hardware in use

- 14 Main menu item "Configuration"
- the SCT can be configured in this menu
- 15 Menu item "use Mini E-MIx" and "use LINMas.." the USB LINMaster or the Mini E-MLX can be chosen



7.4.2. Build process

AVIZOU DET GOIT	the second s			
Nindow Configuration Help	Melexis			
		Small things ma a big difference	ake	En 111/12
nject name & path 2. Project Sel	ings 3.Copy & compile project			
Step 3.1 - Save actual configu	ation			
	1			
16Save Configuration				
Step 3.2 - Build new firmware				
17	Remove objects, Hex- and ELI	F-files ← 18		
Build firmware	Compile MelexCM libraries			
	I▼ Compile Loader	19		
			1	
Step 3.2 · Upload firmware		20		
Step 3.2 · Upload firmware	cl	20		
21 Load Firmware into the	c	20		
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window	c	20		
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step	C 7: compile application	20		
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	C 7: compile application 	- 20	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	7: compile application 111111111111111111111111111111111111	- 20	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	7: compile application 111111111111111111111111111111111111	- 20	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	7: compile application ())))))))))))))))))))))))))))))))))))	- 20 4	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \ [MLX81200 SCT] Step execute - finished [MLX81200 SCT] Step execute	C 7: compile application ())))))))) 6: clean application 5: compile and install Load inished	20 a der	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	7: compile application 7: compile application 6: clean application 5: compile and install Load inished 4: clean Loader sources	- 20 a der	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \ [MLX81200 SCT] Step execute - finished [MLX81200 SCT] Step execute - 1 [MLX81200 SCT] Step	C 7: compile application 111111111111111111111111111111111111	- 20 a der	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	C 7: compile application 111111111111111111111111111111111111	20 a der	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	C 7: compile application 7: compile application 6: clean application 5: compile and install Load inished 4: clean Loader sources 3: compile MelexCM librarie	d der ished	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \	C 7: compile application 111111111111111111111111111111111111	der es ished	22	
Step 3.2 · Upload firmware 21 Load Firmware into the Compile porcess window [MLX81200 SCT] Step execute \ (MLX81200 SCT] Step execute - finished [MLX81200 SCT] Step execute - finished [MLX81200 SCT] Step execute - finished [MLX81200 SCT] Step execute - finished	C 7: compile application 111111111111111111111111111111111111	20 d der es ished ries	22	

Fig. 19 MX81200 SCT – screenshot running compile process of the BLDC-demo kit firmware

- 16 Button "Save configuration"
- 17 Button "Build firmware"
- 18 Checkbox "Remove objects, Hex- and Elf-files"
- 19 Checkbox "Compile MelexCM libraries"
- 20 Checkbox "Compile Loader"
- 21 Button "Load Firmware into the IC"
- 22 Log window "Compile process window"

the configuration will be saved into the files start the compile process remove the old object and firmware files compile the complete MelexCM libraries build the loader firmware from current configuration start the programmer tool and load the actual firmware into the MLX81200 via pin LIN The compile log messages are shown inside



7.4.3. Upload the firmware

- Disconnect all other devices from the LIN bus
- Connect the USB-LIN master version 2.0 or the Mini E-MIx emulator to the pin LIN and to the PC
- Press the Button "Load Firmware into the IC" The programmer tool is started and will be configured for the used loader protocol
- Do a POR of the MLX81200 when the message ">>> Echo : Entering programming mode..." is shown. Uploading the firmware is only possible immediately after POR and for a certain time (500ms). Afterwards uploading is not possible anymore.
- Wait until the E-MIx MM Programmer tool has finished with the message "Info : Programming completed OK"
- Close the E-MIx MM Programmer tool.

A working loader inside the IC is necessary for this process.

	📔 E-Mlx MM Programmer - D:\proje	cts\MLX81200\5of	tware\Borland_CPP\SCT\BLDC_V2.0_	DemoKit_beta\mr	nf\MelexCM_LIN 🚂	
	File Tools Help					
(81200 SCT GUI1	🖃 🧱 MelexCM LIN LDR	Memory Pattern	Sequence Programmer			
Vindow ⊆onfiguration Help Melexis	- 🖧 Flash	Parameter	Value			-
						-
	- C. NyRam	S Width	1 bits			
		Start	Address 0 (0v0)			
lectronic Integrated Systems	C Flash (Fast LIN)	10 Size	32768 (0x8000) words of 1 bits (4.0 KBut	20		
ect name & path 2 Project Settings 3.Cop	W NyBan (Fast LIN)	S Format	Little Endian	~~)		
eer name of part en refeer oorange	- Re InfoPage (Fast LIN)					
Step 3.1 - Save actual configuration	- C. Flash (Standalone)	C Dump	C Program using Edited Data			
1		Program using I	HEX file: C Compare Dump to HEX file :			
Save Configuration		C. Dump and Com		51		
		r Dump and Com				
		D:\projects\MLX8	1200\Software\Borland_CPP\SCT\BLDC_\	/2. 🗾 🔚 👝		
Step 3.2 · Build new firmware		Lise Erase All		23		
I I Re	em	I USE LIGSE All				
Build firmware 🔽 Co	m	Sti	p Executi	en l		
	m (
	- New 100 - 1	(c.				-
Step 3.2 - Upload firmware	Messages Memory View					
1	>>> Echo : FLS: 10 83 D3 CA 2	25 7E				
Load Firmware into the IC	>>> Echo : FLS: 21 26 52 B3 9	94 47 96				
	>>> Echo : FLS: 22 45 94 47 9	96 45 FF				
	>>> BCho : FLS: 23 3B A/ B6 5	2 72 81				
Compile porcess window	>>> Echo : FLS: 25 C7 A7 B6 8	B1 C7 A7				
.ep.data 0x0	>>> Echo : FLS: 26 B7 92 B5 9	90 46 08				
.debug_abbrev 0xd47	>>> Echo : FLS: 27 00 A0 63 0	DB 10 55				
.debug_info 0x2a6e	>>> Echo : FLS: 28 B2 81 A6 0	02 01 8E				
.debug_line 0x33b4	>>> Echo : FLS: 29 B6 49 62 8	38 26 00				
.debug_frame 0xal0	>>> Echo : FLS: ZA 5C 96 46 C	D1 00 70				
.debug_pubnames 0x8b5	>>> MCno : FLS: 28 26 88 26 8	53 BZ 92				
.debug_aranges 0x1d0						
.debug_str 0x16f3				Dunning	A CDC lasked	-
.comment 0x120	1			Running	CKC IOCKED	_
22 	BUILDING OK D:/projects/HLX81200/Software/B	orland_CPP/SCT/	BLDC_V2.0_Dem			
4			<u>)</u>			

Fig. 20 MLX81200 SCT – uploading procedure with the programmer tool

- 21 Button "Load Firmware into the IC"
- 22 Log window "Compile process window"
- 23 E-MIx MM Programmer tool

Start the programmer tool and load the actual firmware into the MLX81200 via pin LIN The compile log messages are shown inside The Melexis programmer tool



8. BLDC DemoKit PC program

8.1. General

The Motor control Software does NOT include any source code for the MLX81200. The required firmware file BLDC_V20_DemoKit_XY.hex MUST be loaded first into the MLX81200. The program MUST be running either by starting the program in the Melexis interactive debugger or let the CPU run free without the emulator out of power on reset.

The BLDC Demokit Firmware can be controlled with the PC program via the LIN bus. The hysteresis brake board can be controlled via the LIN bus too.

For programming a HEX file to the flash please refer to chapter 6.2

8.2. Command line parameter

It is possible to configure the minimum and maximum speed slider values of the graphical interface. The default limits are 500 rpm for the minimum and 5000 rpm for the maximum. Following parameters are supported:

- -maxspeed:x set the maximum limit
- -minspeed:x set the minimum limit

Example: The command:

"MLX81200_Demokit_PC_program.exe -maxspeed:4000 -minspeed:2000"

starts the BLDC DemoKit PC program with a speed range from 2000 to 4000 rpm.



8.3. Main window



Button "Disconnect" 1

disconnect the LIN or CAN master. The application does not close. The software can be reactivated by clicking the 'connect' button. connect the LIN master or the CAN master

- 2 Button "Connect" 3
- The interface can be selected with this two radio buttons
- Interface radio buttons Checkbox "Control Load" open a window to control the hysteresis brake board via the LIN bus
- 5 Button "About"
- 6 Checkbox "Control Debug" open the debug window to control the wave form
- 7 Button "Exit" close all windows and exit the application
- 8 Status bar

4

show the status of the LIN bus and the revision number of the firmware





Fig. 22 Main window with connected LIN master

9 10 11 12 13	Control bar "PWM" Button "Send PWM" Control bar "Speed" Button "Send Speed" Button "Start"	adjust the PWM value (send to MLX81200 with button 10) send the adjusted PWM value to the MLX81200 adjust motor speed (send to MLX81200 with button 12) send the adjusted speed value to the MLX81200 start the motor
14 15	Button "Stop" Status window	stop the motor immediately shows the current motor speed , PWM value , motor status and the current Note: If the CAN interface is used, the value of the current will not be transferred
16	Button "Get Status"	Click this button to get the actual status of the motor (shown in 15) Note: the received values of the PWM and speed will be transferred automatically to the control bars (9 and 11)
17	Checkbox "continuously"	get the motor status and send the target speed value every 150 ms.



8.4. Control Load window



Fig. 23 Control Load window

- 18 Control bar "Load "
- 19 Button "send Load"
- 20 Button "Enable"
- 21 Button "Disable"
- 22 Status window "Current"
- 23 Button "Get status"
- 24 Button "Connect LIN"
- 25 Button "Disconnect"
- 26 Button "Close"
- adjust the load value (send to brake board with button 19) send the adjusted load value to the brake board switch the load on switch the load off shows the actual current of the load click this button to get the actual current of the load (shown in 22)
- connect the LIN master if the CAN interface is used in main window
- disconnect the LIN master
- close the window "Control load" and return to the main window



8.5. Control Debug window



34 Edit field "Parameter2" not used

35 Button "send parameter" not used

8.6. PID controller and PC program

The firmware revision 1.6 for the MLX81200 BLDC DemoKit includes a PID controller.

After the button "Start" is pressed, the start algorithm begins to rotate the motor. After a defined number of motor states (ECM must be active), the PID controller is activated and the predefined target rational speed is set up by the PID controller. Every time the button "Send PWM" is used to change the PWM value of the motor, the PID controller is disabled and only the adjusted PWM is used by the firmware. The PID controller can be reactivated by sending a new speed target to the MLX81200.

9. Appendix

9.1. Schematics of the Evaluation Board

9.2. Schematic of the Evaluation Board – Part 2 Programmer Interface

9.3. Schematic of the Power Board

10. History record

Rev.	No.	Change	Date
1.0	1	Creation of document	20/Dec/06
1.1	2	Changed description from EMLX-MM programmer to Mini E-MLX	13/Feb/08
		and adapted to hardware revision EVB rev1.4 and Powerboard rev1.1	
1.2	3	Add chapter 7 BLDC DemoKit PC program	1/May/08
1.3	4	Add chapter 7 MLX81200 Software configuration tool	24/Okt/08
1.4	5	Add chapter "8.2 Command line parameter", adapted new SW structure	05/Nov/08
		and add chapter "7.4.1 Configure the hardware in use"	

11. Disclaimer

Devices sold by Melexis are covered by the warranty and patent indemnification provisions appearing in its Term of Sale. Melexis makes no warranty, express, statutory, implied, or by description regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. Melexis reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with Melexis for current information. This product is intended for use in normal commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by Melexis for each application.

The information furnished by Melexis is believed to be correct and accurate. However, Melexis shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interrupt of business or indirect, special incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of Melexis' rendering of technical or other services.

© 2008 Melexis NV. All rights reserved ..

For the latest version of this document, go to our website at: www.melexis.com

Or for additional information contact Melexis Direct:

Europe and Japan: Phone: +32 1367 0495 E-mail: sales_europe@melexis.com All other locations: Phone: +1 603 223 2362 E-mail: sales_usa@melexis.com

ISO/TS 16949 and ISO14001 Certified