

Features and Benefits

Compatible with nearly any pc running windows 95, 98 or NT
 Uses the standard parallel port
 No additional supplies required
 Easy to use software included
 Supports all Melexis kitparts using the SPI interface.



Applications

Lab and prototype use.
 Used to program all Melexis devices using the SPI interface

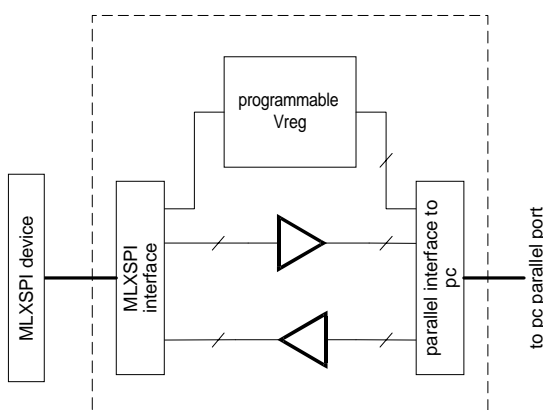
Ordering Information

Part No.	Description
MLXSPI-01	Melexis SPI Programmer unit including software

Accessories

Part No.	Description
MLX90804 kitpart	Kitpart for the MLX90804 speed regulator for universal motors
MLX90805 kitpart	Kitpart for the MLX90805 intelligent triac controller

1. Functional Diagram



2. Description

The MLXSPI programmer is designed to have a flexible and low cost programming tool for all Melexis devices using the SPI interface. The MLXSPI-01 programmer contains its own programmable power supply which can be driven by nearly all standard parallel ports on IBM compatible PC's.

The MLXSPI programmer needs to be controlled with the accompanying software, running under windows 95, windows 98 or windows NT. The PC requires no custom configuration.

TABLE OF CONTENTS

FEATURES AND BENEFITS.....	1
APPLICATIONS	1
1. FUNCTIONAL DIAGRAM	1
2. DESCRIPTION	1
3. GENERAL DESCRIPTION	3
4. COMMUNICATION	3
5. INSTALLATION	3
6. HARDWARE DESCRIPTION	3
7. SOFTWARE DESCRIPTION	3
8. GENERAL INTERFACE	4
9. PARAMETER DEFINITION INTERFACE.....	5
10. MLX90804 DEFINITION INTERFACE.....	6
11. MLX90805 DEFINITION INTERFACE.....	7
12. SPI FILE SYNTAX	8

3. General Description

The MLXSPI-1 programmer was specifically designed to program the Melexis family of kitparts using the SPI interface.

The programmer is a combination of hardware and software. The hardware needs to be connected to the parallel port of a standard pc to operate. There is no need for an additional supply. It is advised that, when connecting the programmer to a pc, no other hardware (like printer, software key, ...) is connected to the same parallel port.

The hardware will generate the correct levels and signals required for reading and writing to the Melexis SPI devices.

The software will generate the timing for all signals, but also gives a user-friendly interface to define all parameters for all different Melexis SPI kitparts. In this way the MLXSPI-01 is a tool which helps to define the behavior of the production device.

4. Communication

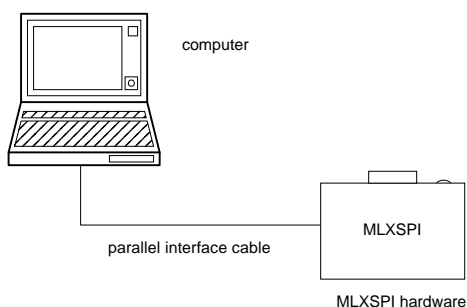
The MLXSPI uses a customized interface to be connected to a parallel port of a pc. Communication in both directions is done, and also the power supply for the MLXSPI programmer and the connected kitpart is taken from the parallel port. It is therefore advised not to connect any other hardware (like printer, software key, ...) to the same parallel port when using the MLXSPI programmer.

5. Installation

The installation is very easy:

First connect the programmer box to your pc via a standard parallel printer cable to a parallel port. Copy the MLXSPI executable to your hard disk. You need 1MB of free disk space.

MLXSPI hardware installation



When running under Windows NT also copy the file IOPORT.SYS to the WINNT\SYSTEM32\DRIVERS directory.

This makes the installation complete.

6. Hardware description

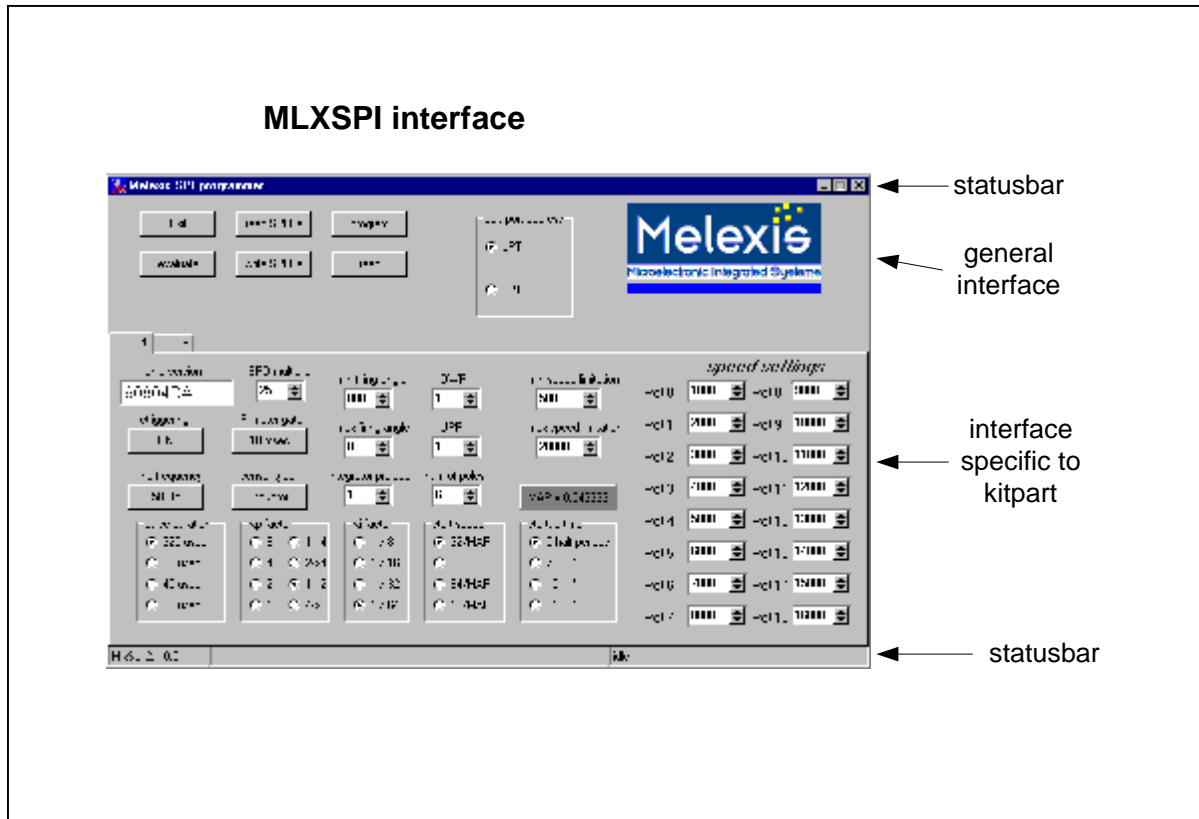
The hardware consists of a box having:

1. A standard connector to connect to a parallel port of a pc.
2. A zif socket to plug in the Melexis SPI device to be programmed. The device must be plugged in such that pin 1 of the chip is at the side of the handle of the socket.
3. A led. When the box is connected to the pc, the software is running and succeeded in making connection with the box, this led is blinking as long as the software is in idle mode. While accessing the SPI kitpart (for reading or writing) the led stops blinking and is led continuously.

7. Software description

When starting the MLXSPI software an interface window is activated on the screen. This interface window consists of:

1. Title bar
The title bar indicates the program name and has a system menu.
2. General interface
This part of the interface contains the command buttons. Following commands are available:
read SPI file
write SPI file
evaluate parameters
program parameters to kitpart
read parameters from kitpart
exit program
select parallel port
3. General interface
Parameter definition interface.
This part of the program interface is specific to the different kitpart versions and therefore consists of several pages, which can be selected by clicking the tabs at the top side.
4. Status bar.
The status bar indicates the software version, the source of the parameters that were imported most recently, and the program status.



8. General interface

The general interface contains the command buttons. The execution of most of the commands depends on the kitpart version that is selected in the parameter definition interface. Following commands are available:

1. Read SPI file:

This command will read all parameters of the kitpart type as selected in the parameter definition interface from the selected file. This file must have been written previously with the 'write SPI file' command. As the file is a normal ascii file, it can also be edited manually. The syntax of the SPI file is explained at the end of this document. The values as read from the file are directly reflected in the parameter definition interface. Parameters that are not present in the file are not updated. The status bar indicates the file name that has been read most recently.

2. Write SPI file:

This command will write all parameters, as they are present in the parameter definition interface to the selected file. If the file already exists, the command will write into this file, eventually overwriting the parameters stored before in this file for the same kitpart. This

means that, if a file contains a set of parameters for MLX90804 and for MLX90805, only that part will be overwritten that corresponds to the version in the parameter definition interface. The status bar indicates the file name that has been written most recently.

3. Program:

This command will send the parameters as defined in the parameter definition interface to the MLXSPI programmer. This will be done in several steps:

At first a *consistency check* is done on the parameters. This check verifies that the set of parameter values that is selected is possible to program, or that the chosen combination does not introduce an unpredictable behavior of the kitpart. For more information see the section describing the parameter definition interface.

Then all parameters are read and transformed to values to be programmed in the kitpart. Most of the times this will introduce rounding errors because not all parameters can be programmed at the resolution as present in the parameter definition interface. For more information see the section describing the parameter definition interface.

The calculated values are now sent to the MLXSPI programmer to be written into the kitpart.

At the end all parameters are read back from the kitpart to verify the programming cycle. Note that the system does not check whether the correct version of the kitpart is plugged into the MLXSPI programmer.

4. **Read:**

The read command will read the values back from the kitpart, will recalculate and display the corresponding parameters in the parameter definition interface.

The status bar indicates that the displayed parameters correspond to the kitpart.

5. **Evaluate:**

The evaluate command is very similar to the combination of the program and read command, but it skips the connection to the MLXSPI programmer.

The command starts with a consistency check, it transforms the parameters to programming values and then recalculates and redisplay the corresponding parameters in the parameter definition interface.

The result corresponds exactly to the values that should have been programmed and shows the result of the rounding process.

6. **Exit:**

This command exits the program.

7. **Select parallel port:**

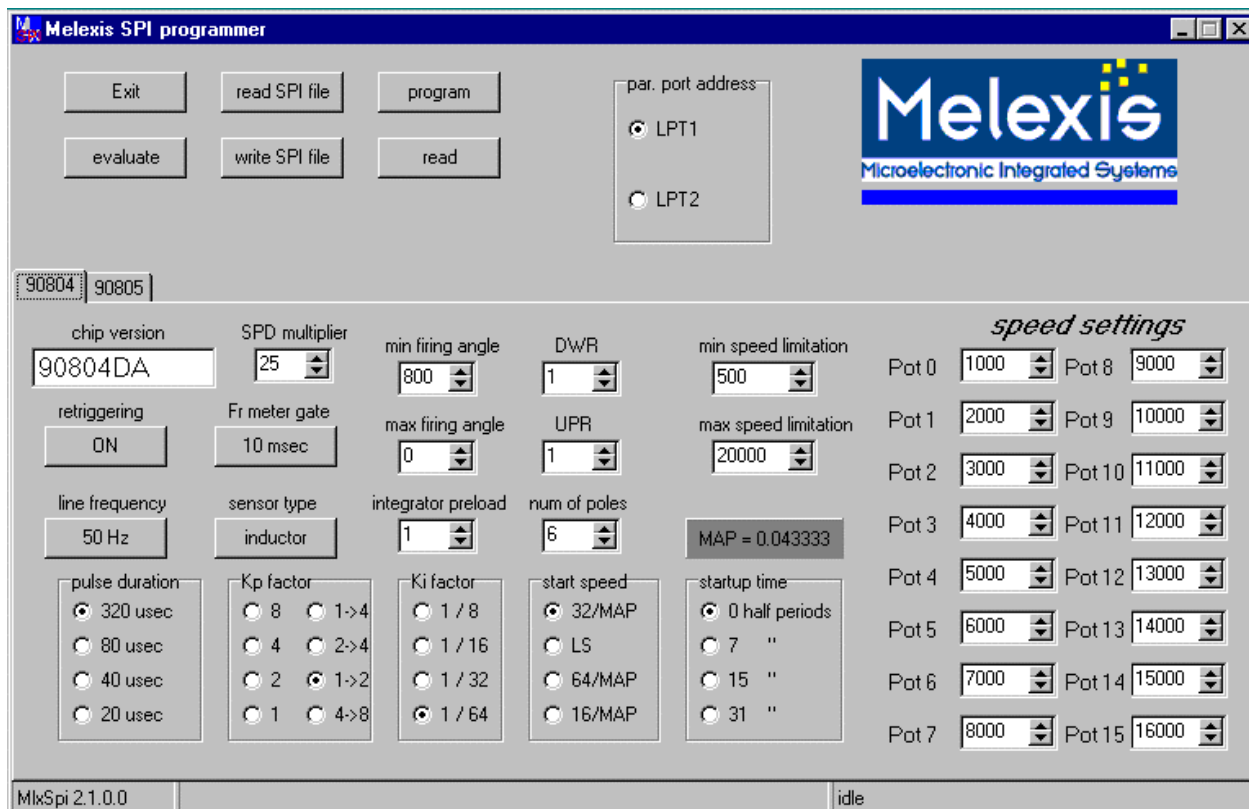
This command gives the software the possibility to connect to parallel port LPT1 or LPT2.

9. Parameter definition interface

This part of the program interface is specific to the different kitpart versions and therefore consists of several pages, which can be selected by clicking the tabs at the top side.

In this interface the values of all parameters are defined. It is therefore the interface where the behavior of the programmed kitpart is created.

The following sections will explain how to use the parameter definition interface for the different types of kitparts. It will however not explain in detail how to define the parameters to obtain a certain behavior. For this the datasheets and application notes corresponding to the production version of the kitpart should be consulted.



10. MLX90804 definition interface

This interface contains all parameters as are described in application note MLX90804AN3 'manual for definition of the options'. Use this document as a guide to define all values. Note that all speed settings are in RPM. The unit for the minimum and maximum firing angle is 0.01*msec. This means that a value of 800 corresponds to 8 msec.

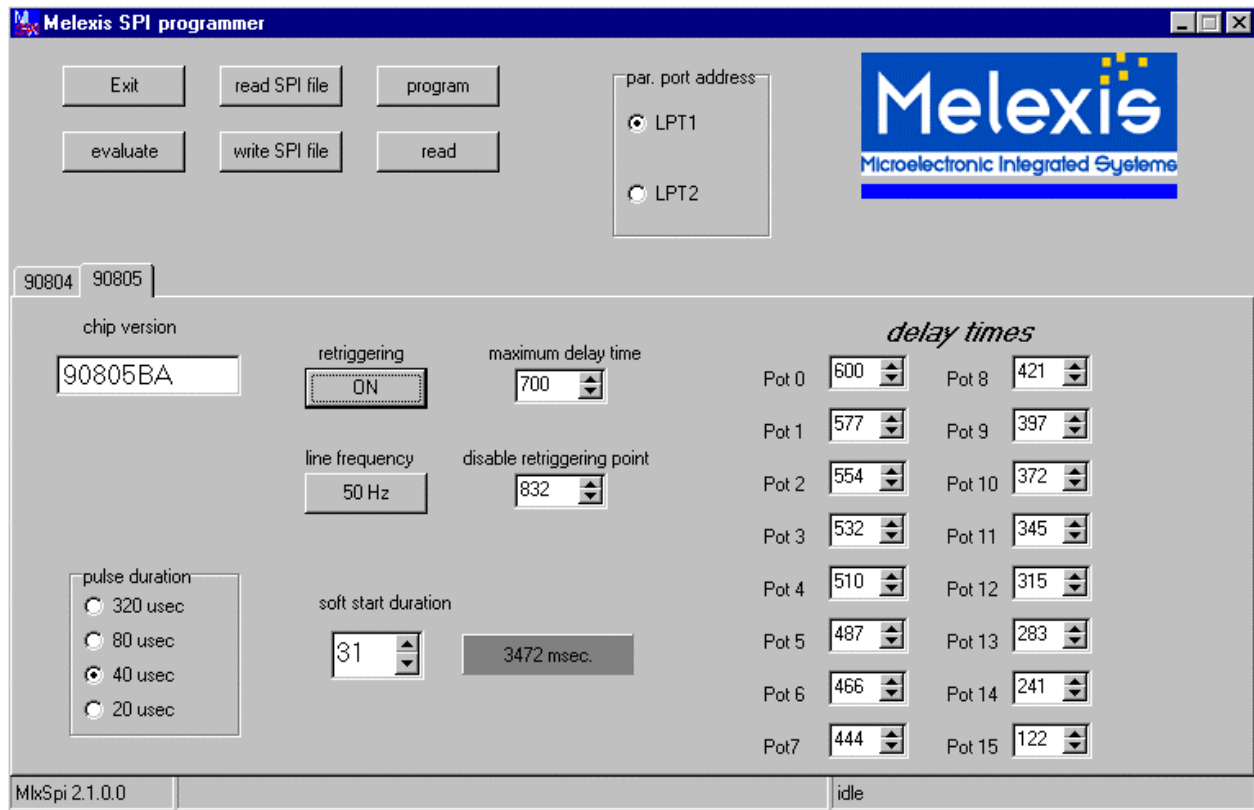
The parameter MAP is calculated and updated automatically by the program.

During the consistency check a lot of verifications are done. A few examples are:

1. Are the low speed limitation, the number of poles and the SPD multiplier coefficient consistent with the requirement as described in MLX90804AN3?
2. Is the maximum speed limitation higher then the speed settings on all potentiometer settings?

During the transformation of the parameters to programming values rounding errors can be introduced. As an example, the maximum speed limitation can only be programmed with a resolution of 5 bits.

The box, indicating 'chip version', is used to distinguish possible future versions of the kitpart. The value can currently not be changed.



11. MLX90805 definition interface

This interface contains all parameters as are described in datasheet MLX90805 'intelligent triac controller'. Use this document as a guide to define all values.

Note that the unit for all delay settings is 0.01*msec. This means that a value of 800 corresponds to 8 msec.

The soft start duration is defined by a value between 0 and 31, but the program automatically calculates and displays the corresponding value in msec.

During the consistency check a lot of verifications are done. A few examples are:

1. Is the phase delay defined for each potentiometer setting smaller then the maximum delay time?
2. Is the phase delay defined for each potentiometer setting smaller then half the period of the line frequency?

During the transformation of the parameters to programming values small rounding errors can be introduced.

The box, indicating 'chip version', is used to distinguish possible future versions of the kitpart. The value can currently not be changed.

12. SPI file syntax

The syntax of the SPI file is very similar to the syntax used in the initialization files used by the operating systems.

The file can contain several groups, with each group having a number of records. The groups correspond to the kitpart type, and each record contains a parameter. Following is a dump of an SPI file containing the default values of all parameters for both MLX90804 and 90805.

```
[ 90804 ]
P04Coil=True
P04FG10M=True
F0450=True
P04RETRIG=True
NumPoles=6
SLIMA[ 4:0 ]=20000
SLIM[ 9:0 ]=500
DUTS[ 1:0 ]=0
SSE[ 1:0 ]=3
PX[ 3:1 ]=6
SFT[ 4:0 ]=0
LD[ 9:0 ]=0
MAX[ 9:0 ]=0
MIN[ 9:0 ]=800
SPE[ 1:0 ]=0
UPR[ 4:0 ]=1
DWR[ 4:0 ]=1
MULT[ 7:0 ]=25
SpeedSetting[ 15 ]=16000
SpeedSetting[ 14 ]=15000
SpeedSetting[ 13 ]=14000
SpeedSetting[ 12 ]=13000
SpeedSetting[ 11 ]=12000
SpeedSetting[ 10 ]=11000
SpeedSetting[ 9 ]=10000
SpeedSetting[ 8 ]=9000
SpeedSetting[ 7 ]=8000
SpeedSetting[ 6 ]=7000
SpeedSetting[ 5 ]=6000
SpeedSetting[ 4 ]=5000
SpeedSetting[ 3 ]=4000
SpeedSetting[ 2 ]=3000
SpeedSetting[ 1 ]=2000
SpeedSetting[ 0 ]=1000
```

```
[ 90805 ]
F0550=True
P05RETRIG=True
DUTS[ 1:0 ]=2
ATN[ 4:0 ]=31
MINA[ 3:0 ]=13
MIN[ 9:0 ]=700
PhaseDelay[ 15 ]=122
PhaseDelay[ 14 ]=241
PhaseDelay[ 13 ]=283
PhaseDelay[ 12 ]=315
PhaseDelay[ 11 ]=345
PhaseDelay[ 10 ]=372
PhaseDelay[ 9 ]=397
PhaseDelay[ 8 ]=421
PhaseDelay[ 7 ]=444
PhaseDelay[ 6 ]=466
PhaseDelay[ 5 ]=487
PhaseDelay[ 4 ]=510
PhaseDelay[ 3 ]=532
PhaseDelay[ 2 ]=554
PhaseDelay[ 1 ]=577
PhaseDelay[ 0 ]=600
```


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:	All other locations:
Phone: +32 13 67 04 95	Phone: +1 603 223 2362
E-mail: sales_europe@melexis.com	E-mail: sales_usa@melexis.com

QS9000, VDA6.1 and ISO14001 Certified