LED-Warrior14U-DR

USB to IEC62386 Bridge

Code Mercenaries

1. Features

- IEC62386 type II master
- USB to IEC62386 bridge
- Receives and transmits all 8, 16, and 24 bit telegrams
- According to DIN EN 62386-103/2011
- 5 V supply via USB
- Galvanically isolated between IEC62386 and USB

1.1 Variants

LED-Warrior14U-DR is available as a DIN rail mountable device.

1.2 Custom variants

Custom variants are possible.

2. Functional overview

LED-Warrior14U-DR is a type II IEC62386 master. It can coexist with other masters on the same bus but is not addressable.

LED-Warrior14U-DR supports transmission of 8 bit reply telegrams, all 16 and 24 bit forward telegrams and receives 8 bit reply telegrams as well as any 16 and 24 bit forward telegrams transmitted by other masters. This allows to control light levels, retrieve status information and set configuration data in IEC62386 devices.

The USB interface is using our IO-Warrior24 generic USB I/O controller. IO-Warrior24 does not need any special drivers to be installed. It does work with standard system drivers on Windows and MacOS. The driver for Linux is part of the standard kernel distribution.

3. Connections

The USB connection is a mini-B USB connector. It is low speed USB compatible backwards including USB 1.1.

The IEC62386 bus is connected to screw terminal blocks. There are two positions for each of the two lines to allow easy feed through (the two positions in either terminal block are the same).

A IEC62386 power supply needs to be added externally if necessary.

3.1 Mechanical dimensions

Dimensions in mm
36 x 90 x 58 (69 max.)

Conforms to standard size for a 2 unit DIN rail mount.

4. Communication with LW14U-DR

LED-Warrior14U-DR is a combination of our IO-Warrior24 and LED-Warrior14 chips in a ready to use unit.
IO-Warrior24 is a generic IO controller with multiple functions. Its I2C function is being used to access the LED-Warrior14. Since none of the other functions of IO-Warriro24 is being used in the LED-Warrior14U-DR the following description will be limited to the I2C function.

4.1. Basic USB access

IO-Warrior24 identifies as a generic HID class device. This has the advantage that on most system platforms a generic driver is available. Access to the IO-Warrior24 is possible with standard file io commands.
Support libraries for use with Windows, Linux, and MacOS are available as part of the IO-Warrior SDK. Documentation for the use of the libraries is contained in the SDK.
There are two logical interfaces on the IO-Warrior24. Only the second interface is used for LED-Warrior14U-DR to access the I2C function.
Data packets are being send to and received from the interface 1 to talk to the I2C function. The exact mechanisms depend on the operating system in use. Please refer to our library code and the IO-Warrior documentation for low level details.
Communication with the IOW24 works by sending and receiving data packets with 8 bytes. The first byte (ReportID) identifies the command type or return data packet type (some operating systems handle this as a 7 byte data packet plus separate ReportID).

4.2 Initializing LW14U-DR

To start communication the first thing is to enable the I2C function of IOW24.
A report with ID = $01 has to be send to do this:

ReportID	1	2	3	4	5	6	7
$01 out	$01	$00	$00	$00	$00	$00	$00

There will be no reply to this command, it just configures the IO-Warrior24 to use I2C.

4.3 Writing data to LW14U-DR

Writing to a register of the LW14 is done by a report with ID = $02:

ReportID	1	2	3	4	5	6	7
$02 out	$Cx	$40	reg#	data	data	$00	$00

$Cx - x is the number of bytes in the transaction = data bytes plus 2 for I2C address and register#
$40 - is the default I2C address of the LW14 (see 4.10)
reg# - is the number of the LW14 register to write to (see 4.5)
data - one or two bytes of data to write to the LW14

Any write transactions are acknowledged by a report with the result of the transfer:

ReportID	1	2	3	4	5	6	7
$02 in	flags	$00	$00	$00	$00	$00	$00

flags contains the following bits:
7 - Error bit, 1=error
6 - unused, zero
5 - unused, zero
4 - unused, zero
3 - unused, zero
2 - data count MSB
1 - data count
0 - data count LSB

Normally the LW14U-DR should never report an I2C error unless a register has been tried to access with too many data bytes.

4.4 Reading data off the LW14U-DR

Reading data from a specific register of the LW14U-DR requires two consecutive transactions. The first is a write transaction to set the register number to read from:

ReportID	1	2	3	4	5	6	7
$02 out	$C2	$40	reg#	$00	$00	$00	$00

The second transaction does the actual reading of data from the LW14 register:

ReportID	1	2	3	4	5	6	7
$03 out	count	$41	$00	$00	$00	$00	$00

count - number of bytes to read.

Data is returned in a report with ID $03:

ReportID	1	2	3	4	5	6	7
$03 in	flags	data	data	data	data	data	data

flags contains the following bits:
7 - error, set if slave does not ack command byte
6 - unused, zero
5 - unused, zero
4 - unused, zero
3 - unused, zero
2 - data count MSB
1 - data count
0 - data count LSB

4.5 LW14 registers

Communication with the LW14U-DR is done via the LW14 registers.
The register number is always reset to zero at the end of a transaction, so reading without first writing a register address always returns the content of the status register.

Register	R/W	Function	Data
$00	R	Status	1 Byte
$01	R/W	Command	3 Bytes
$02	W	Config	1 Byte
$F0	R	Signature	6 Bytes
$FE	W	Set Addr	2 Bytes

4.6 Status register

The status register is one byte that contains the bus status and command status flags:
7 - Bus Error Status, 0 = Bus OK, 1 = Bus fault
6 - Busy, 0 = ready, 1 = busy
5 - Overrun
4 - Frame Error
3 - Valid Reply
2 - Reply Timeframe, < 22 Te since last command
1 - MSB byte count for telegram received
0 - LSB byte count for telegram received

Bus Error Status = 1 indicates that the bus is not working, either another device is pulling it permanently low or the bus is not connected. Commands to register 1 will be ignored if the bus is not working.
Busy = 1 indicates that the last command has not yet been transmitted. Any new command sent to register 1 will be ignored until the last command has been transmitted and the busy bit is cleared.
Overrun = 1 is set if a new telegram is received before the last one was read from register $01. This bit is reset be reading register $01.
Frame Error = 1 if an invalid telegram has been on the bus since last read of the status register. Reset by reading the status register.
Valid Reply = 1 if a telegram has been received within 22 Te (1 Te = 1/2 bit cell on IEC62386 = 416μs) of sending a command. If the received telegram is a forward telegram from another master this indicates that the gap between forward telegrams has been violated. Reset by reading register $01.
Reply Timeframe = 1 indicates that the time frame for a reply from the last addressed device has not yet timed out. This bit is set to 1 after the transimission of a command and is reset to zero after 22 Te or on bus activity.
The two lowest bits indicate the number of bytes in the last received telegram. 0 indicates there is no new data, 1 is a 1 byte reply telegram, 2 and 3 indicate 16 and 24 bit forward telegrams respectively. Both bits are reset upon reading register $01.
Versions of LW14 prior to V2.0.0.0 did use these two bits as flags to indicate 8 or 16 bit telegrams. The use as a counter is upwards compatible, but care must be taken in existing software to check that if both bits are set this is decoded properly.

4.7 Command register

The command register accepts one to three bytes of data. Writing a single byte to the command register causes the LW14 to immediately send this byte as a return telegram onto the IEC62386 bus. No checking of bus status is done, so this may cause a bus collision as is required by some IEC62386 interactions. Inter telegram timing has to be taken care for by the application, LW14 transmits the telegram immediately after receiving it via I2C.
Writing two or three bytes causes a 16 or 24 bit IEC62386 command to be send onto the bus. The first byte is the address byte, the second (and third) the command byte.
LW14 does observe bus avctivity and intra telegram timing when sending a 16 or 24 bit telegram. Priotrity of the telegram is set via the config register (see 4.8).

4.7.1 IEC62386 Commands

LW14 receives all 16 and 24 bit IEC62386 forward telegrams.
Please refer to the IEC62386 specification for details on the commands.

4.8 Config register

The Config register contains configuration options for the IEC62386 bus.

7 - unused, write zero
6 - unused, write zero
5 - unused, write zero
4 - unused, write zero
3 - unused, write zero
2 - Bus priority MSB
1 - Bus priority
0 - Bus priority LSB

Bus priority sets the priority for the commands to be transmitted. Valid values are 1 to 5, values will be clipped if out of range. Default value is 2.
Use priority 1 for commands within a transaction, except for the first command. Priority 2 is for user issued commands, 3 for the start of a multi-command transaction, 4 for automatically generated commands, 5 for commands and starts of transactions that query status or memory.

4.9 Signature register

The signature register can be used to identify LED-Warrior14 and get the revision information for the chips firmware. The content of the signature is fixed and can not be changed. It contains 6 bytes with the following content:
0 - VendorID MSB
1 - VendorID LSB
2 - ProductID MSB
3 - ProductID LSB
4 - Version MSB
5 - Version LSB

The 16 bit VendorID allows us to differentiate standard and custom chips. Standard chips use 0 as our ID.
Product is a 16 bit product code, LED-Warrior14 has 14 as its product code value.
Version is the four digit BCD version number identifying the chips firmware version. I.e. V1.0.3.5 would be stored as $1035.

4.10 Set Address register

With the Set Address register it is possible to move LW14 to a different I2C address.
To prevent address reprogramming by mistake the address has to be send in normal and inverted format to register $FE. The address is transmitted in 7 bit right aligned format (i.e. values range from 1 to 127).
Values of 128 and more are not accepted.

The first byte has to contain the address in normal format (i.e. values 0 to 127), the second byte must contain the value of the first byte XORed with $FF.

Using this function does usually make no sense for the LED-Warrior14U-DR variant. It is inteded to be used when designing with the chips or modules to connect multiple LED-Warrior14 to one host. In case of the LED-Warrior14U-DR multiple units can be distinguished by their USB connection.

5. Absolute maximum ratings

Supply voltage (Vcc relative to GND):	-0.5V to +7V
Input voltage into USB pins (relative to GND):	GND - 0.5V to Vcc + 0.5V
IEC62386 input voltage (differential):	max. 50V
IEC62386 current draw	max. 5mA
Storage temperature:	-55°C to +100°C
ESD:	2000V human body model

Absolute maximum ratings must not be exceeded or permanent damage to the LED-Warrior14U-DR may result.

5.1 Operating specifications

Supply voltage (USB V+ relative to GND):	4.5V to 5.25V
Operating temperature:	0°C to +70°C
Supply current:	30mA max.
IEC62386 input voltage (differential):	max. 24V
IEC62386 input current:	max. 2mA

6. Ordering information

Partname	Order Code	Package	MOQ	Description
LED-Warrior14U-DR	LW14U-DR	DIN rail	1	IEC62386 master with USB

The chips and modules listed here are standard products. Customized chips and modules are available on request.

6.1 Packaging info

The modules are packaged as single units in cardboard boxes.

6.2 Shipping version

LED-Warrior14U-DR is currently shipping in version V2.0.0.0

6.2.1 Revision History

V2.0.0.0 - Added handling 24 bit telegrams.

V1.2.0.0 - Added transmitting 8 bit telegrams.

V1.1.0.0 - Initial shipping version.

6.3 FCC / CE

LED-Warrior14U-DR does confrom to CE. The declaration of CE conformity is availabel online. FCC has not been applied for.

Legal Stuff

The information contained herein is subject to change without notice. Code Mercenaries makes no claims as to the completeness or correctness of the information contained in this document.

Code Mercenaries assumes no responsibility for the use of any circuitry other than circuitry embodied in a Code Mercenaries product. Nor does it convey or imply any license under patent or other rights.

Code Mercenaries products may not be used in any medical apparatus or other technical products that are critical for the functioning of lifesaving or supporting systems. We define these systems as such that in the case of failure may lead to the death or injury of a person. Incorporation in such a system requires the explicit written permission of the president of Code Mercenaries.

Trademarks used in this document are properties of their respective owners.

Code Mercenaries
Hard- und Software GmbH
Karl-Marx-Str. 147a
12529 Schönefeld
Germany
Tel: +49-3379-20509-20
Mail: support@codemercs.com
Web: www.codemercs.com

HRB 9868 CB
Geschäftsführer: Guido Körber, Christian Lucht