RS485 I2C Adapter V3.3

Manual

Coptonix GmbH

Luxemburger Str. 31
D – 13353 Berlin
Phone: +49 – (0)30 – 61 74 12 48
Fax: +49 – (0)30 – 61 74 12 47
www.coptonix.com

1. Introduction

The *RS485 I2C adapter* is an universal applicable I2C-tool with a 128 bytes of buffer and an adjustable SCL-frequency up to 400 kHz. With the RS485 I2C adapter as master numerous bus participants can be adressed such as IOExpander, sensors, LCDs, 7-segment display, stepping motors, AD/DA converters, real time clocks, tone generators, RAM, EEProms, etc.

The SCL frequency can be adjusted between 12 - 400 kHz.

The RS485-address of the adapter is adjustable via an 8bit DIP-switch. Thus several adapters could be attached at a RS485 bus.

The adapter contains an I2C level shifter on board. Thus, it is possible to connect the adapter to an I2C bus having different voltage levels between 2V and 15V.

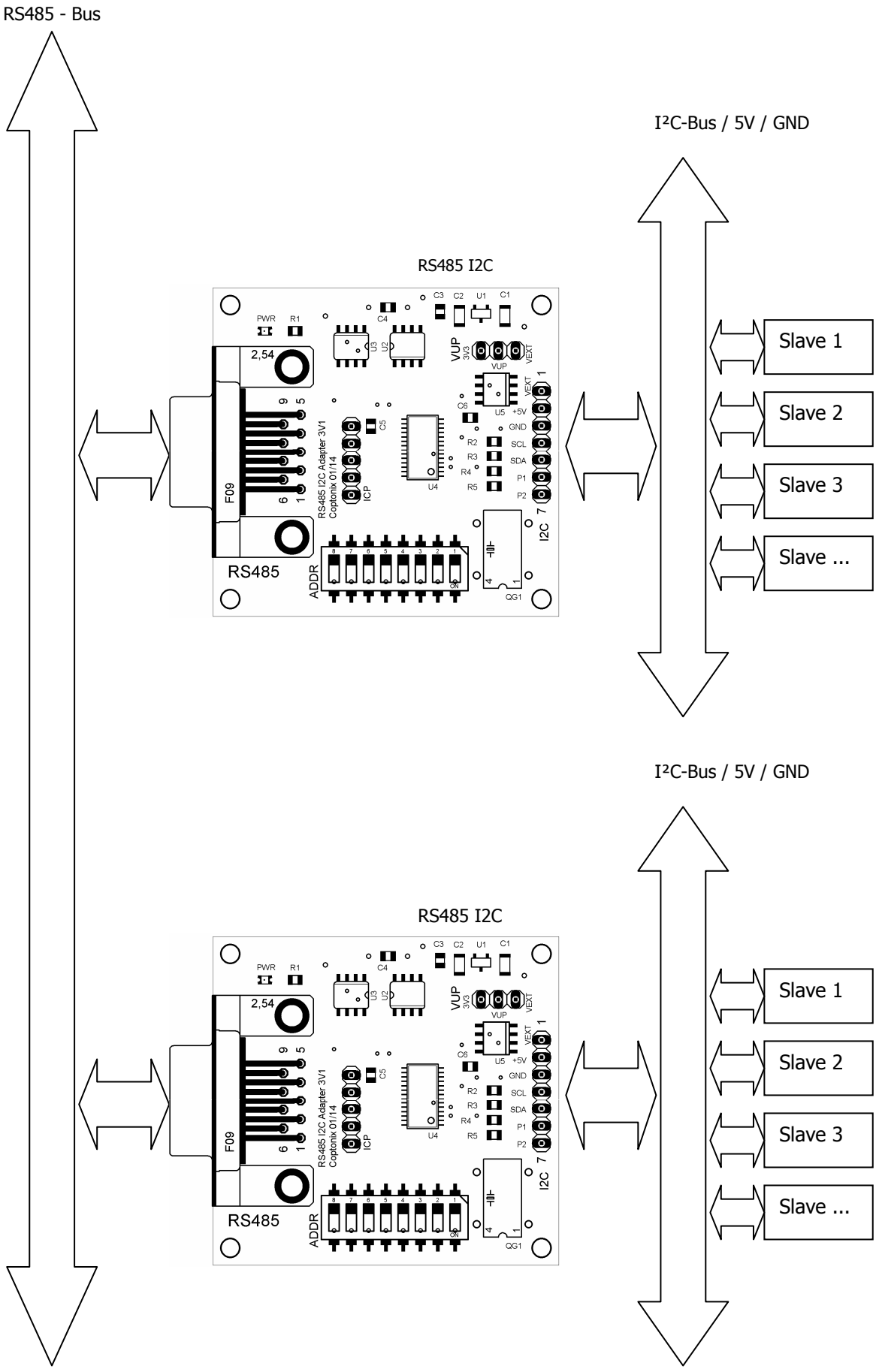
Labview VIs and a 32bit DLL (for Windows) are included in delivery. This makes the integration of the adapter into own applications possible.

It is possible to communicate with the adapter using Windows API functions such as CreateFile(), WriteFile() and ReadFile(). A simple software interface (ASCII commands) is available.

Some of the software Tools are for the developer very helpfully. Thus it is possible to test immediately I2C devices. The software "IIC Control" supports EEPROMS of 1kbit (128 bytes) to 1Mbit (128k byte).

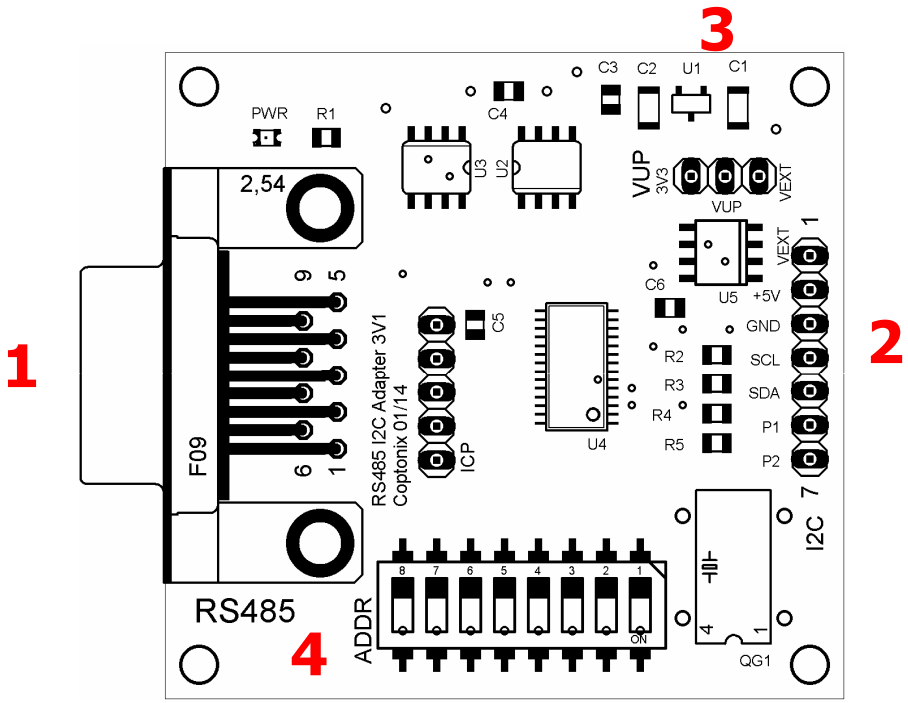
<u>Features:</u>

- Configurable I2C frequency 12Hz - 400kHz
- adjustable duty cycle (SCL-frequency)
- On board I2C level shifter, I2C levels from 2V to 15V
- supports multi-master
- Master transmit & receive
- supports clock streching
- 7bit addressing
- 2 digital general inputs/outputs
- Simple software interface / ASCII commands
- 32Bit DLL for Windows
- Labview VIs


RS485 - Bus
I²C-Bus / 5V / GND
RS485 I2C
Slave 1
Slave 2
Slave 3
Slave ...
RS485
ADDR
I2C
I²C-Bus / 5V / GND
RS485 I2C
Slave 1
Slave 2
Slave 3
Slave ...
RS485
ADDR
I2C

2. Interface

- **1** RS485-Interface
  Pin1 : **B** (RX)
  Pin2 : **A** (RX)
  Pin3 : **Z** (TX)
  Pin4 : **Y** (TX)
  Pin5 : Ground
  Pin6 – 9: not connected.
- **2** I²C-Interface
  Pin 1: External pull-up voltage V_{ext} (2V – 15V)
  Pin 2: Supply voltage +5V
  Pin 3: Ground
  Pin 4: I2C – SCL
  Pin 5: I2C – SDA
  Pin 6: IO-Pin 1
  Pin 7: IO-Pin 2
  Header strip / 2.54 pitch
- **3** Jumper VUP – $V_{pull-up}$ Pull-up voltage
  Pin 1: +3.3V
  Pin 2: $V_{pull-up}$; connected to pull-up resistors(4K7)
  Pin 3: External pull-up voltage
- **4** S2 (DIP-Switch)
  Default addresse 0xFF.



3. Characteristics *(only Version 3.3 and later)*

	Min.	Typ	Max.	Unit
Power-Supply				
Supply Voltage		5.0		V
Supply Current		12	15	mA
I2C-Bus pins (SCL, SDA)				
V_{ext} External Pull-up Voltage	2	-	15	V
V_{IH} High-State Input Voltage	$0.58V_{pull-up}$	-	-	V
V_{IL} Low-State Input Voltage	-	-	$0.42V_{pull-up}$	V
Limiting values				
Interrupt pins				
Input Voltage	0	-	5.5	V
Output Voltage	0	-	$V_{DD(3,3V)}$	V
Power-Supply				
Supply Voltage	**4.0**	**5.0**	**6.0**	**V**
Temperature				
operating temperature	-20	-	+70	°C

4. Software interface

Function	CMD + Data (->RS485)	Feedback (<- RS485)	
WriteI2C	**DS+'77'+SA+' XXYYZZ....'+CS**+<CR>	**'77'+DS+SA+'01'+CS**+<CR>	Data write OK
		'77'+DS+SA+'00'+CS+<CR>	Slave not found
ReadI2C	**DS+'72'+SA+Cnt+CS**+<CR>	**'72'+DS+SA+'01'+ CS**+<CR>	Slave found
		'64'+DS+SA+ Cnt+'XXYYZZ....'+CS+<CR>	Data read OK
		'72'+DS+SA+'00'+ CS+<CR>	Slave not found
CheckSlvAdr	**DS+'63'+SA+CS**<CR>	**'63'+DS+SA+'01'+CS**+<CR>	Slave found
		'63'+DS+SA+'00'+CS+<CR>	Slave not found
SetSCLFreq	**DS+'65'+IH+IL+CS**+<CR>	**'65'+DS+IH+IL+CS**+<CR>	SCL-Freq. Write OK
GetSCLFreq	**DS+'69'+CS**+<CR>	**'69'+DS+IH+IL+CS**+<CR>	SCL-Freq. Read OK
ReSetIO1	**DS+'6D01'+CS**+<CR>	**'6D'+DS+'01'+CS**+<CR>	IO-Pin 1 HIGH
	DS+'6D00'+CS+<CR>	**'6D'+DS+'00'+CS**+<CR>	IO-Pin 1 LOW
ReSetIO2	**DS+'6E01'+CS**+<CR>	**'6E'+DS+'01'+CS**+<CR>	IO-Pin 2 HIGH
	DS+'6E00'+CS+<CR>	**'6E'+DS+'00'+CS**+<CR>	IO-Pin 2 LOW
GetIOState	**DS+'6F'+CS**+<CR>	**'6F'+ DS+00+CS**+<CR>	IO1 LOW – IO2 LOW
		'6F'+ DS+01+CS+<CR>	IO1 HIGH – IO2 LOW
		'6F'+ DS+02+CS+<CR>	IO1 LOW – IO2 HIGH
		'6F'+ DS+03+CS+<CR>	IO1 HIGH – IO2 HIGH

Further Messages	Description	
ChkSumERROR	**'73'+DS+'01'+CS**+<CR>	CheckSum Error
UnCMD	**'FF'+DS+'00'+CS**+<CR>	Unknown command

DS: DIPSwitch, Adapter's Address
SA: Slave Address
CS: CheckSum
CS = 0x0100 – (Sum MOD 0x0100); Sum is the sum of all bytes without CS and CR.
Cnt: Count of bytes to read (s. ReadI2C)

IH: SCL-Frequenz - High
IL: SCL-Frequenz - Low
f_{SCL} = 12.000.000 / (2 * (IL + IH)) ; if **IL** and **IH** are equal, then the duty cycle is 50%.
XXYYZZ...: data to write / read.
<CR> : CarriageReturn (0x0D). Commands and data are always terminated with a CarriageReturn.

XXYYZZ... data to write / read. At least 1 byte and maximally 128 bytes may be transferred
Example: the 5 bytes 0xA1, 0x1F, 0x22, 0x5C, 0xB0 are to be sent to the adapter address 0xFE to the slave address 0xC4. Then the following string (terminated with a carriage return) is sent over the serial interface:

'**FE77C4A11F225CB0DB**'+**<CR>**

'FE' → 0xFE; Adapter address, DIP-Switch
'77' → 0x77; Command / WRITE
'C4' → 0xC4; Slave address
'A11F225CB0' → Data: 0xA1, 0x1F, 0x22, 0x5C, 0xB0
'59' → 0x59; Checksum of '**FE77C4A11F225CB0**'
Sum = 0x46+0x45+0x37+0x37+0x43+0x34+0x41+0x31+0x31+0x46+0x32+0x32+0x35+0x43+0x42+0x30 = 0x03A7
(F) (E) (7) (7) (C) (4) (A) (1) (1) (F) (2) (2) (5) (C) (B) (0)
CS = 0x0100 – (Sum MOD 0x0100) = 0x0100 – (0x03A7 MOD 0x0100) = 0x0100 – 0xA7 = 0x59
<CR> → 0x0D; CarriageReturn

The feedback from the adapter may be as follow:
'**77FEC4012F**'+**<CR>** or '**77FEC40030**'+**<CR>**

'77' → 0x77; Command
'FE' → 0xFE; Adapter address ; DIP-Switch
'C4' → 0xC4; Slave address
'01' → 0x01; Slave address found, data written successfully.
'2F' → 0x2F; Checksum of '**FE77C401**'
<CR> → 0x0D; CarriageReturn

or

'77' → 0x77; Command
'FE' → 0xFE; Adapter address ; DIP-Switch
'C4' → 0xC4; Slave address
'00' → 0x01; Slave address not found, data write aborted.
'30' → 0x30; Checksum of '**FE77C400**'
<CR> → 0x0D; CarriageReturn

RS485 – settings:

Baud:	19200
DataBits:	8
StopBits:	1
Parity:	None