

RS232 I2C Adapter Manual

Rev. 8

Coptonix GmbH

Falkentaler Steig 9

D – 13467 Berlin

Phone: +49 – (0)30 – 61 74 12 48

Fax: +49 – (0)30 – 61 74 12 47

www.coptonix.com

1. Introduction

The *RS232 I2C Adapter* is an universal applicable I2C-tool with a 128 bytes of buffer and a SCL-frequency up to 400 kHz. With the *RS232 I2C Adapter* as master numerous bus participants can be addressed such as IOExpander, sensors, LCDs, 7-segment display, stepping motors, AD/DA converters, real time clocks, tone generators, RAM, EEPROMs, etc.

The Tool is ideally for the developer, who would like to develop and/or test own I2C circuits. The SCL frequency can be adjusted between 1kHz - 400 kHz. The frequency depends on the Adapter's type.

Following are the different types we offer:

- **200kHz:** SCL-frequency adjustable between 8kHz and 200kHz. Needs external power supply of 5V.
- **400kHz:** SCL-frequency adjustable between 12kHz and 400kHz. Needs external power supply of 5V.
- **LowPower:** SCL-frequency adjustable between 1kHz and 100kHz. No separate power packs for voltage supply are needed. Power supply is provided from the RS232- Interface (DTR & RTS). Typical power consumption of about **5mA**.

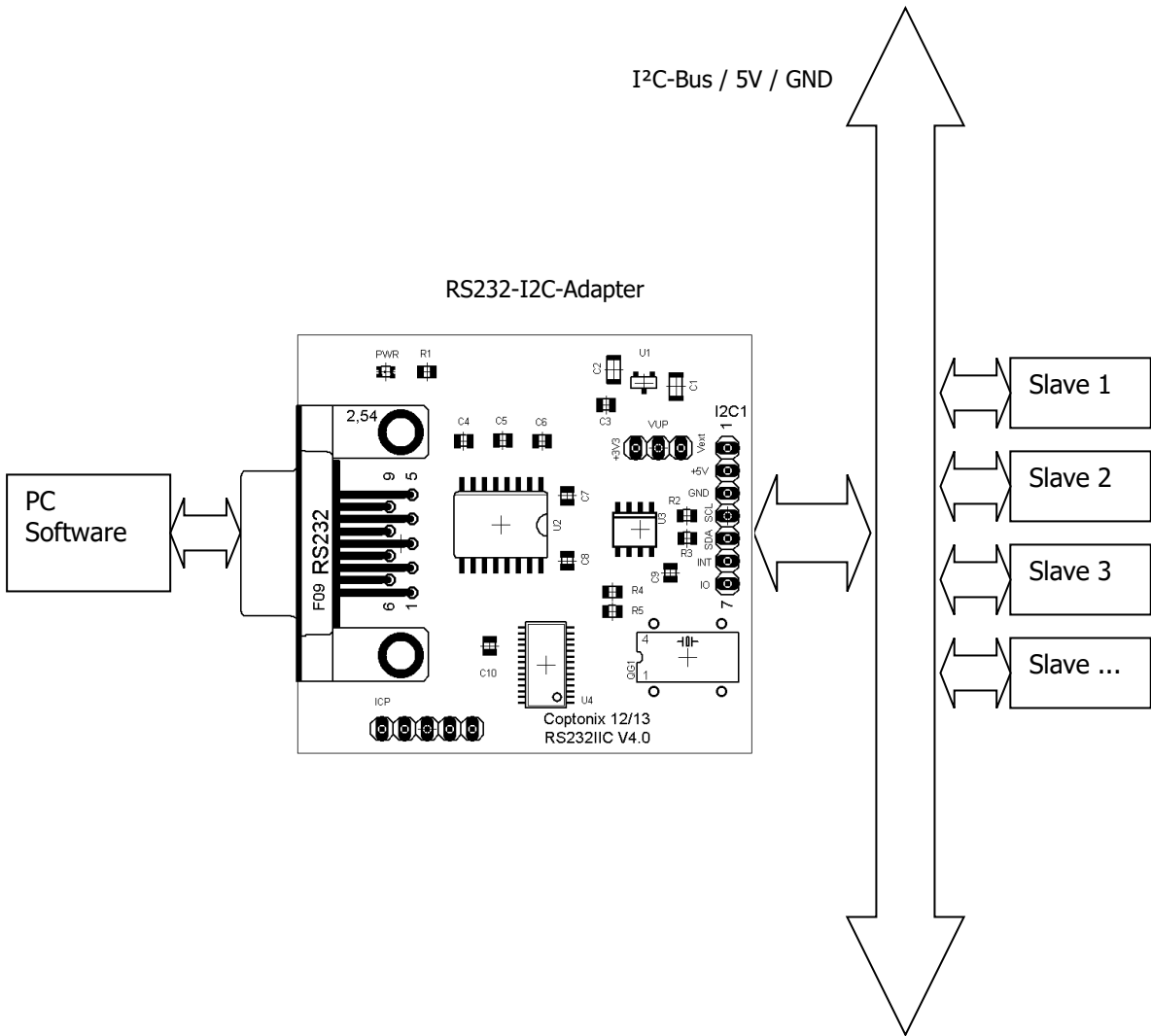
The adapter (only 200 & 400 types) contains an I2C level shifter on board. Thus, it is possible to connect the adapter to an I2C bus having different voltage levels between 2.5V and 15V.

The software interface allows communication without drivers or other software. 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).

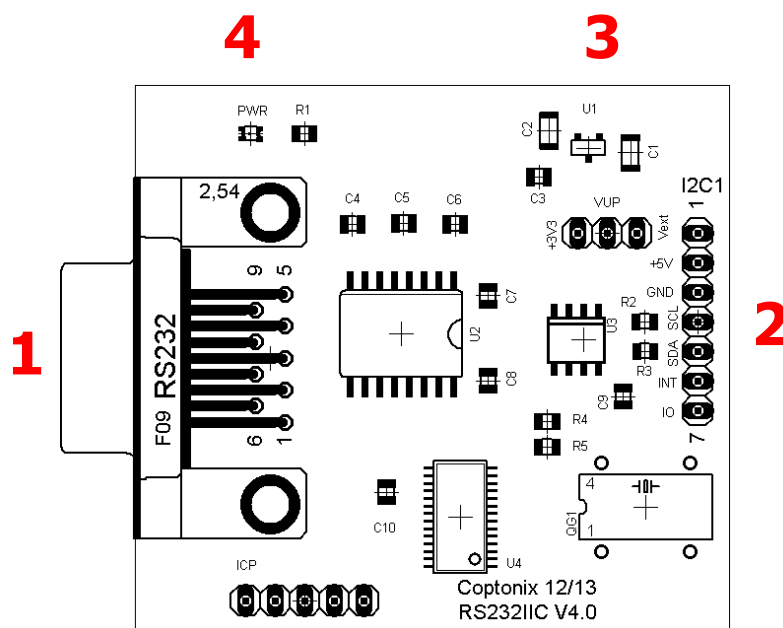
Features:

- RS232 Sub-D
- Configurable I2C frequency 1kHz - 400kHz
- adjustable duty cycle (SCL-frequency)
- On board I2C level shifter, I2C levels from 2.5V to 15V
- supports multi-master
- Master transmit & receive
- supports clock stretching
- 7bit addressing
- Interrupt input for external events
- Simple software interface / ASCII commands
- Labview VIs



2. Interface (200kHz und 400kHz Adapter)

- **1** RS232 port for communication with a PC (RxD, TxD, GND)
- **2** I²C-Interface
 Pin 1: External pull-up voltage V_{ext} (2.5V – 15V)
 Pin 2: Supply voltage +5V
 Pin 3: Ground
 Pin 4: I2C – SCL
 Pin 5: I2C – SDA
 Pin 6: Interrupt – Input 1
 Pin 7: Not used
 Header strip / 2.54 pitch
- **3** Jumper VUP – $V_{\text{pull-up}}$ Pull-up voltage
 Pin 1: +3.3V
 Pin 2: $V_{\text{pull-up}}$; connected to pull-up resistors (4K7)
 Pin 3: External pull-up voltage
 Position 1-2: connects pull-up voltage to internal +3.3V
 Position 3-2: connects pull-up voltage to external pull-up voltage.
- **4** LED Power-ON



Interface (LowPower Adapter)

- **1** D-SUB 9 **female**
RS232-Interface
Pin 1: Not connected
Pin 2: TxD
Pin 3: RxD
Pin 4: DTR (power supply)
Pin 5: GND
Pin 6: Not connected
Pin 7: RTS (power supply)
Pin 8: Not connected
Pin 9: Not connected

- **2** D-SUB 9 **male**
I²C-Interface
Pin 1: GND
Pin 2: I2C – Clock (SCL)
Pin 3: I2C – Data (SDA)
Pin 4: Interrupt-Input
Pin 5: Not connected
Pin 6: Not connected
Pin 7: Not connected
Pin 8: Not connected
Pin 9: Not connected

3. Characteristics

(200kHz und 400kHz Adapter)

	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	2.5	-	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	0	-	+70	°C

(LowPower Adapter)

	Min.	Typ	Max.	Unit
Power-Supply				
Supply Voltage		RS232-powered		V
Supply Current		4	5	mA
I2C-Bus pins (5V tolerant I/O pins)				
V _{IH} High-State Input Voltage	0.7V _{DD(3,3V)}	-	-	V
V _{IL} Low-State Input Voltage	-	-	0.3V _{DD(3,3V)}	V
Limiting values				
I/O pins (SCL, SDA, Interrupt)				
Input Voltage	0	-	5.5	V
Output Voltage	0	-	V _{DD(3,3V)}	V
Power-Supply				
Supply Voltage	4.5	-	24.0	V
Temperature				
operating temperature	0	-	+70	°C

4. Software interface

Function	Code			Description		
	Hex	Char	Parameter	CMD + Data (->RS232)	answer (<- RS232)	
Write I2C	0x77	w	SlvAddr, data (d ₁ .. d ₁₂₈), CheckSum	'w'+SA+'XXYY..' +CS+<CR>	'77'+01'+SA+CS+<CR>	successfully written
					'77'+00'+SA+CS+<CR>	read error
Read I2C ¹	0x72	r	SlvAddr, Cnt (Number of bytes to read) , CheckSum	'r'+SA+Cnt+CS+<CR>	'72'+01'+SA+CS+<CR>	successfully read
					'64'+SA+'XXYY....'+Cnt+CS+<CR>	data packet
Read I2C ²	0x52	R	SlvAddr, Cnt (Number of bytes to read) , CheckSum	'R'+SA+Cnt+CS+<CR>	'72'+00'+SA+CS+<CR>	read error
					'52'+01'+SA+'XXYY....'+Cnt+CS+<CR>	successfully read
Write Read I2C ² (Repeated Start)	0x57	W	SlvAddr, Cnt (Number of bytes to read), data (d ₁ .. d ₁₂₈) , CheckSum	'W'+SA+Cnt+'XXYY..' +CS+<CR>	'52'+00'+SA+CS+<CR>	read error
					'57'+01'+SA+'XXYY....'+Cnt+CS+<CR>	successfully read
					'57'+10'+SA+CS+<CR>	write error
					'57'+20'+SA+CS+<CR>	read error
CheckSlvAdr	0x63	c	SlvAddr, CheckSum	'c'+SA+CS+<CR>	'57'+30'+SA+CS+<CR>	unkown error
					'63'+SA+01'+CS+<CR>	slave connected
ScanI2C ²	0x43	C	CheckSum	'C'+CS+<CR>	'63'+SA+00'+CS+<CR>	slave not connected
					'43'+Cnt+'XXYYZZ....'+CS+<CR>	Cnt devices connected
SetSCLFreq ₁	0x65	e	SCLH, SCLL, CheckSum	'e'+IH+IL+CS+<CR>	'43'+00'+CS+<CR>	Zero devices connected
					'65'+IH+IL+CS+<CR>	set frequency
GetSCLFreq ₁	0x69	i	CheckSum	'i'+CS+<CR>	'69'+IH+IL+CS+<CR>	read frequency
SetSCLFreq ₂	0x45	E	32Bit SCL Freq. (low byte first), CheckSum	'E'+LL+LH+HL+HH+CS+<CR> Low.....High	'45'+LL+LH+HL+HH+CS+<CR> Low.....High	set frequency
GetSCLFreq ₂	0x49	I	CheckSum	'I'+CS+<CR>	'49'+LL+LH+HL+HH+CS+<CR> Low.....High	read frequency
ReSetIRQ	0x71	q	Interrupt state (active)	'q01'+CS+<CR>	'71'+01'+CS+<CR>	interrupt active
			Interrupt state (deactive)	'q00'+CS+<CR>	'71'+00'+CS+<CR>	interrupt deactive

1) Deprecated commands

2) Commands available from firmware version 6V0 and later.

IRQ	'70'+ '01' +CS+<CR>	interrupt detected
ChkSumERROR	'73'+ '01' +CS+<CR>	Checksum Error
UnCMD	'FF'+ '00' +CS+<CR>	unknown command

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

SCL Frequency (LL LH HL HH):

Example:

Set SCL frequency of 100kHz (100000 = 0x000186A0).

The frequency is transmitted low byte first: '**45A0860100**'+CS+<CR>

XXYY...: data to be send / read.

<CR> : CarriageReturn (0x0D). Commands and data are always terminated with a CarriageReturn.

XXYY... data to be send. 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 address 0xC4. Then the following string (terminated with a carriage return) is sent via the serial interface: '**wC4A11F225CB0DB**'+<CR>

'w' → 0x77; Command / WRITE

'C4' → 0xC4; Slave Address

'**A11F225CB0**' → Data: 0xA1, 0x1F, 0x22, 0x5C, 0xB0

'**DB**' → 0xDB; Checksum for '**wC4A11F225CB0**'

Sum = 0x77 + 0x43 + 0x34 + 0x41 + 0x31 + 0x31 + 0x46 + 0x32 + 0x32 + 0x35 + 0x43 + 0x42 + 0x30 = 0x0325

(w) (C) (4) (A) (1) (1) (F) (2) (2) (5) (C) (B) (0)

CS = 0x0100 – (Sum MOD 0x0100) = 0x0100 – (0x0325 MOD 0x0100) = 0x0100 – 0x25 = 0xDB

<CR> → 0x0D; CarriageReturn

The feedback from the adapter can be the following:

'7701C4BA'+<CR> or '**'7700C4BB'**+<CR>

'77' → 0x77; Command

'01' → 0x01; Slave address found, data written successfully.

'C4' → 0xC4; Slave address

'**BA**' → 0xBA; Checksum calculated for '**'7701C4**'

<CR> → 0x0D; CarriageReturn

or

'77' → 0x77; Command
'00' → 0x01; Slave address not found. Communication aborted.
'C4' → 0xC4; Slave address
'BB' → 0xBB; Checksum calculated for **'7700C4'**
<CR> → 0x0D; CarriageReturn

RS232 – Settings:

Baud: 19200 (ask for other baud rate)
DataBits: 8
StopBits: 1
Parity: None