RS232 I2C Adapter V4.0

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 *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 adressed 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 15 - 400 kHz. The frequency depends on the Adapter's type.

Following are the different types we offer:

- **200kHz**: SCL-frequency adjustable between 8 and 200kHz. Needs external power supply of 5V.
- **400kHz**: SCL-frequency adjustable between 12 and 400kHz. Needs external power supply of 5V.
- **LowPower**: SCL-frequency adjustable between 1 and 100kHz. No separate power packs for voltage supply are needed. Power supply is provided from the RS232- Interface (DTR & RTS). Power consumption less than **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 2V and 15V.

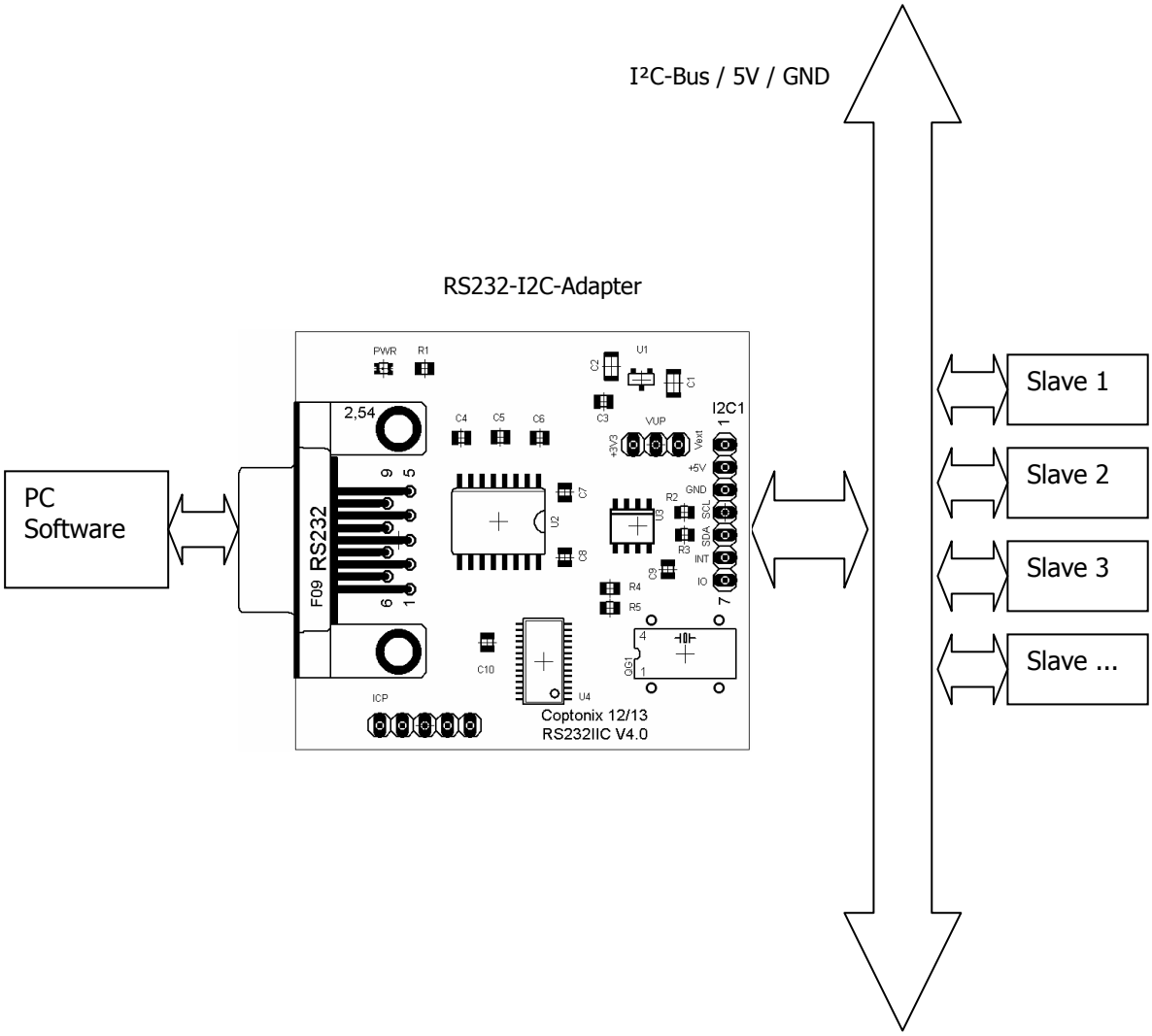
Labview VIs and a 32bit DLL (for Windows) are included in delivery. This makes the integration of the adapter into own applications possible.

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).

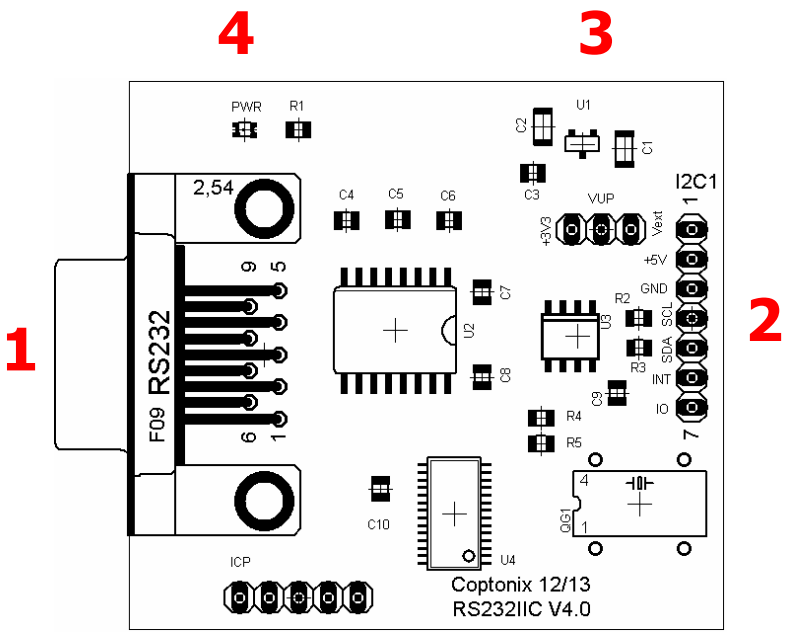
Features:

- RS232 Sub-D
- Configurable I2C frequency 8Hz - 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
- Interrupt input for external events
- Simple software interface / ASCII commands
- 32Bit DLL for Windows
- Labview VIs


I²C-Bus / 5V / GND
RS232-I2C-Adapter
PC
Software
RS232
F09
2,54
PWR
R1
U1
C1
C2
C3
C4
C5
C6
VUP
I2C1
+3V3
Vext
+5V
GND
SCL
SDA
INT
IO
R2
R3
R4
R5
C7
C8
C9
C10
U2
U3
U4
QG1
ICP
Coptonix 12/13
RS232IIC V4.0
Slave 1
Slave 2
Slave 3
Slave ...

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} (2V – 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 – $V_{pull-up}$ Pull-up voltage
  Pin 1: +3.3V
  Pin 2: $V_{pull-up}$; connected to pull-up resistors (4K7)
  Pin 3: External pull-up voltage
  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	-	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

(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	-20	-	+70	°C

4. Software interface

Function	Code			Description		
	Hex	Char	Parameter	CMD + Data (->RS232)	answer (<- RS232)	
WriteI2C	0x77	w	SlvAdr,d_1,d_2,…d_{128},CheckSum	**'w'**+**SA**+**' XXYY….'**+**CS**+<CR>	**'77'+'01'+SA+CS+**<CR>	Success
					'77'+'00'+SA+CS+<CR>	Write Error
ReadI2C	0x72	r	SlvAdr,Cnt,CheckSum	**'r'**+**SA**+**Cnt**+**CS**+<CR>	**'72'+'01'+SA+CS**+<CR> **'64'+SA+'XXYY….'+Cnt+CS**+<CR>	Sucess Data packet
					'72'+'00'+SA+CS+<CR>	Read Error
CheckSlvAdr	0x63	c	SlvAdr,CheckSum	**'c'**+**SA**+**CS**<CR>	**'63'+SA+'01'+CS**+<CR>	Slave address
					'63'+SA+'00'+CS+<CR>	Slave not found
SetSCLFreq	0x65	e	I2SCLH,I2SCLL,CheckSum	**'e'+IH+IL+CS**+<CR>	**'65'+IH+IL+CS**+<CR>	Set frequency
GetSCLFreq	0x69	i	CheckSum	**'i'+CS**+<CR>	**'69'+IH+IL+CS**+<CR>	Read frequency
ReSetIRQ	0x71	q	State (aktiv)	**'q01'+CS**+<CR>	**'71'+'01'+CS**+<CR>	Interrupt active
			State (deaktiv)	**'q00'+CS**+<CR>	**'71'+'00'+CS**+<CR>	Interrupt deactive

	Description	
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
IH: SCL-Frequency - High
IL: SCL-Frequency – Low ; if **IL** and **IH** are equal, then the duty cycle is 50%.
f_{SCL} = 7.372.800 / (2 * (IL + IH)) ; *200kHz - Adapter*
f_{SCL} = 12.000.000 / (2 * (IL + IH)) ; *400kHz - Adapter*
f_{SCL} = 1.000.000 / (2 * (IL + IH)) ; *LowPower - Adapter*

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