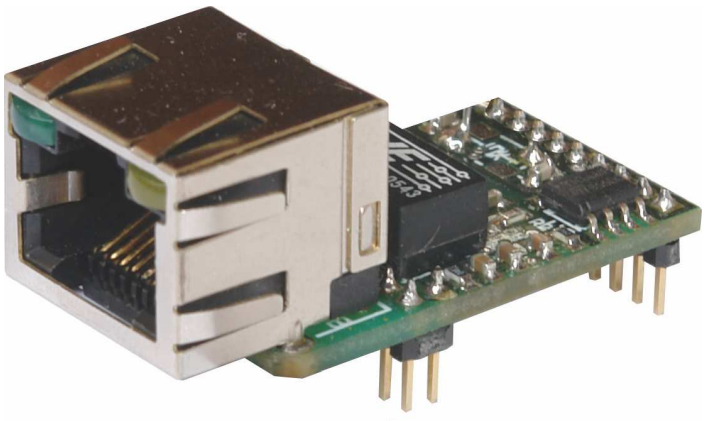
XT-NANO



DesignGuide V1.0

1 Objective

This Design-Guide documents how the XT-NANO can be integrated into customer systems. This documentation is a recommendation to the best of our knowledge. AK-NORD does not assume any liability for the information in this documentation nor for any damages related to or caused by the use of this Design Guide.

2 Product description

XT-NANO combines an asynchronous serial interface with TTL-level. It has several bus systems as RS232, RS485, I2C and SPI. The highest transmission speed becomes at the SPI - interface achieves. With 12,5 MHz approx. 200 KByte or 2.000.000 MBit can be transmitted effectively with TCP/IP. At the SPI- and I2C-BUS XT-NANO functions only in MASTER-MODE

3 Software interfaces

3.1 Connect procedures

The following dial procedures are supported:

- AT commands
- Pad commands
- Connect-On-Data
- I2C MasterMode
- SPI MasterMode

3.2 AT commands

Via AT-Commands you may control the LAN connection and change the configuration of XT-NANO

3.2.1 Configuration commands

A range of parameters can be controlled by configuration commands of XT-NANO as listed below:

- Setting IP – Address and the Port
- Setting Gateway and Subnetmask
-

3.2.2 AT connection commands

Command	Function	Response
ATDxxx	Establishes a LAN connection	Connect
ATH	Disconnects the LAN Connection.	OK
DTR Drop	Disconnects the LAN Connection.	

Whenever the LAN connection with a communication partner is established, a transparent channel for serial data is provided.

3.3 PAD – commands

Via PAD-Commands you may control the LAN connection.

3.3.1 PAD connection commands

Command	Function	Response
XXX,Userdata	Establishes a LAN connection	Connect
DTR Drop	Disconnects the LAN Connection.	

3.4 Connect-On-Data

You can setup via Telent or Browser the Connection. If the XT-NANO receives any Data, the connection established automatically

3.5 Firmware updates

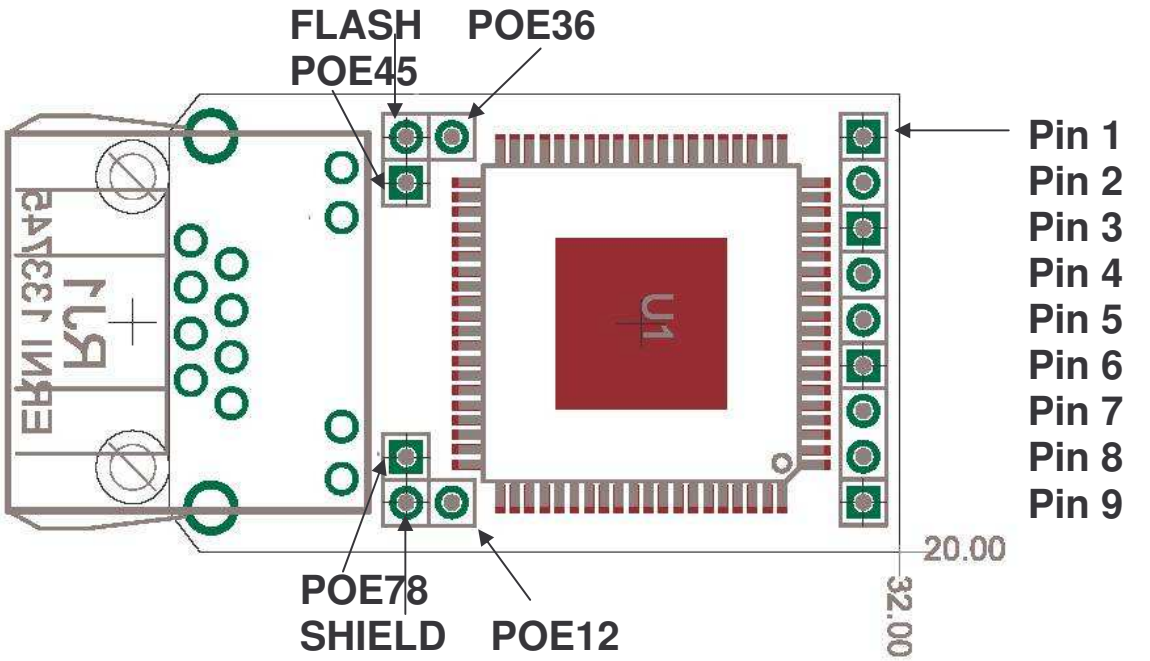
Firmware updates can send via TCP/IP. You need the Software XT-ADMIN

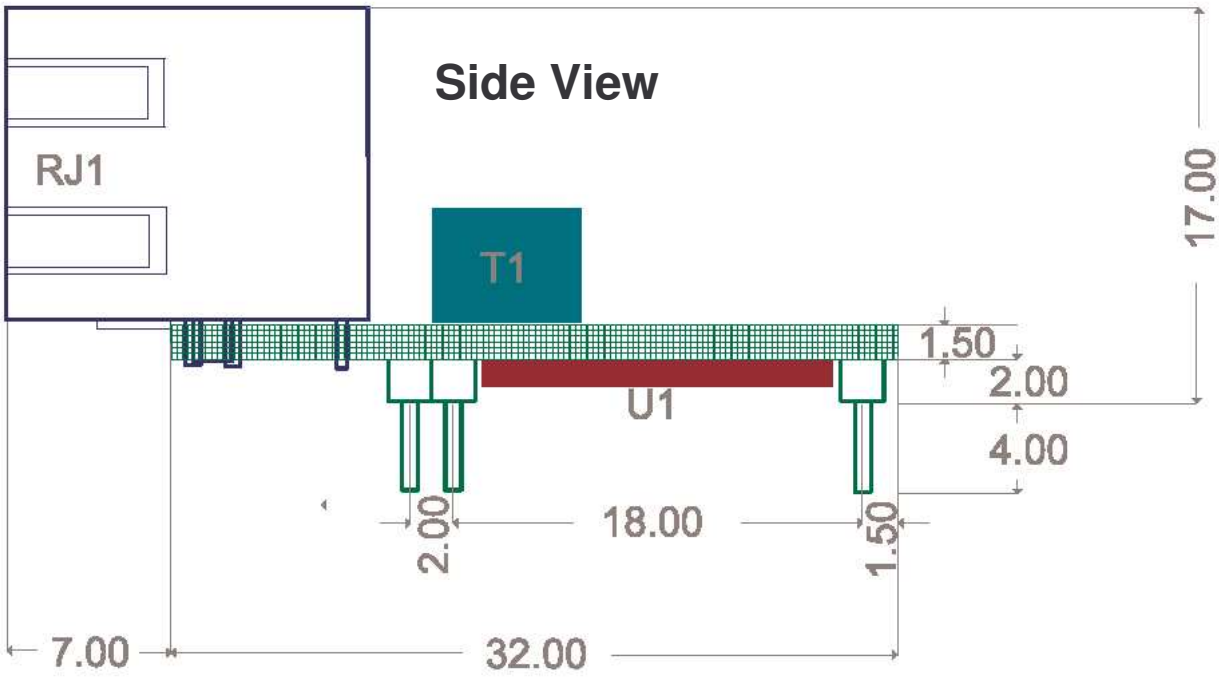
3.6 Supported network protocols

1. IP
2. TCP
3. UDP
4. FTP
5. ICMP
6. ARP
7. SNMP
8. DHCP
9. BOOTP
10. DNS
11. DDNS
12. TELNET
13. HTML
14. http
15. DYNDNS

4 Dimensions of the XT-NANO

Bottom View





4.1 Pin - Description

PIN	RS232		RS485		I2C		SPI	
1	RXD	←	RXD	←			MOSI	→
2	TXD	→	TXD	→			MISO	←
3	DSR	←					SCK	→
4	DTR	→	RW	→			SS\	→
5	RTS	→			SCL	→		
6	CTS	←			SDA	← →		
7	Reset\	←	Reset\	←	Reset\	←	Reset\	←
8	+3.3Volt	←	+3.3Volt	←	+3.3Volt	←	+3.3Volt	←
9	Ground	←	Ground	←	Ground	←	Ground	←
POE12		→	Connect to middle of PIN 1 and PIN 2 (RJ45)					
Shield		→	Connect to RJ45 → Shield					
POE78		→	Connect to RJ45 → PIN 7 and PIN 8					
POE45		→	Connect to RJ45 → PIN 4 and PIN 5					
Flash			Don't use					
POE36		→	Connect to middle of PIN 3 and PIN 6 (RJ45)					

→ OUT
← IN

5 Hardware interfaces of the XT- NANO

The XT-NANO is connected via the 2mm Pins. This includes:

- Power supply
- Power over Ethernet POE
- V.24/RS-232 with TTL level (3.3V)
- RS485 with TTL level (3.3V)
- Inter Integrated Circuit (IIC) BUS with TTL level (3.3V)
- Serial Peripheral Interface (SPI) with TTL level (3.3V)

5.1 Power Supply

- 3,3 V +- 5% low noise, the supply voltage is directly used (0 Ohm)

5.1.1 Power consumption and power down modes

The following values are approximate power consumption values in the different states:

Condition	Power consumption
100Mbit LAN Speed	300 mA
10Mbit LAN Speed	180 mA

5.2 Ethernet interface

Can directly used without additional components. RJ45 Connector on board.

10 Half Duplex
10 Full Duplex
100 Half Duplex
100 Full Duplex
AutoSensing

5.3 RS232 interface

L = BUS: N Use RS232 - BUS

XT-NANO interface has an asynchronous serial interface with TTL level.

- Transmission speeds 300 – 115200 bps (asynchronous)
- Character representation:
  7 bit / 8 bit
  no, Even, Odd ,Mark, Space parity
  1, 2 stop bit
- Flow control hardware (RTS/CTS), Software (XON/XOFF)

5.4 RS485 interface

L = BUS: R Use RS485 - BUS

XT-NANO interface has an asynchronous serial interface with TTL level. It's for and two wire connection with read/write switching.

- Transmission speeds 300 – 115200 bps (asynchronous)
- Character representation:
  7 bit / 8 bit
  no, Even, Odd ,Mark, Space parity
  1, 2 stop bit
- Flow control Software (XON/XOFF)

5.5 I2C Interface

L = BUS: I Use I2C – BUS

XT-NANO interface has a 2-wire I2C interface with TTL level. Only MASTER mode is supported

- Transmission speeds up to 100KHz
- Character representation: 8 bit
- Flow control : switchable

5.6 SPI Interface

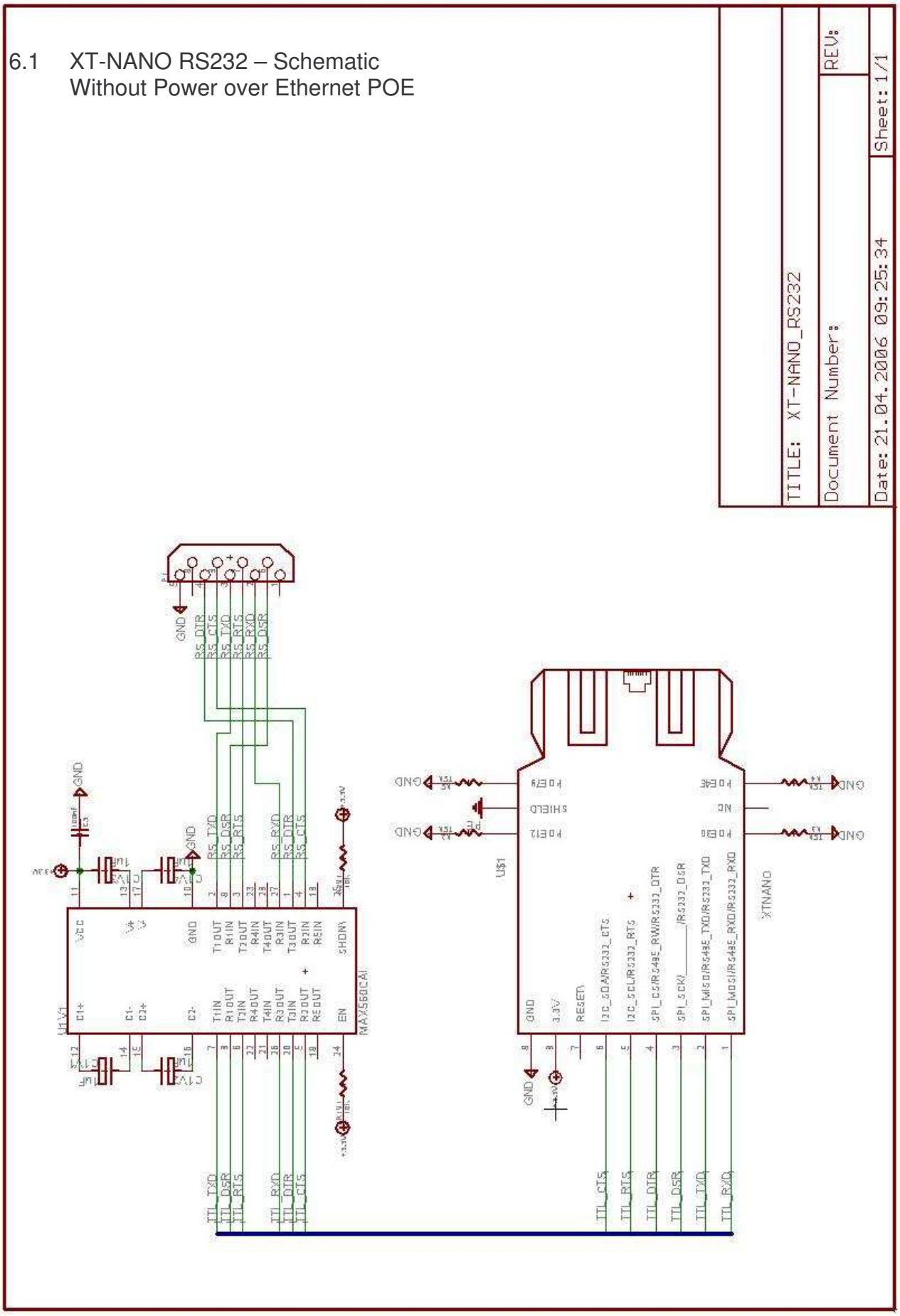
L = BUS:S Use SPI – BUS

XT-NANO interface has a 4-wire SPI interface with TTL level. Only MASTER mode is supported

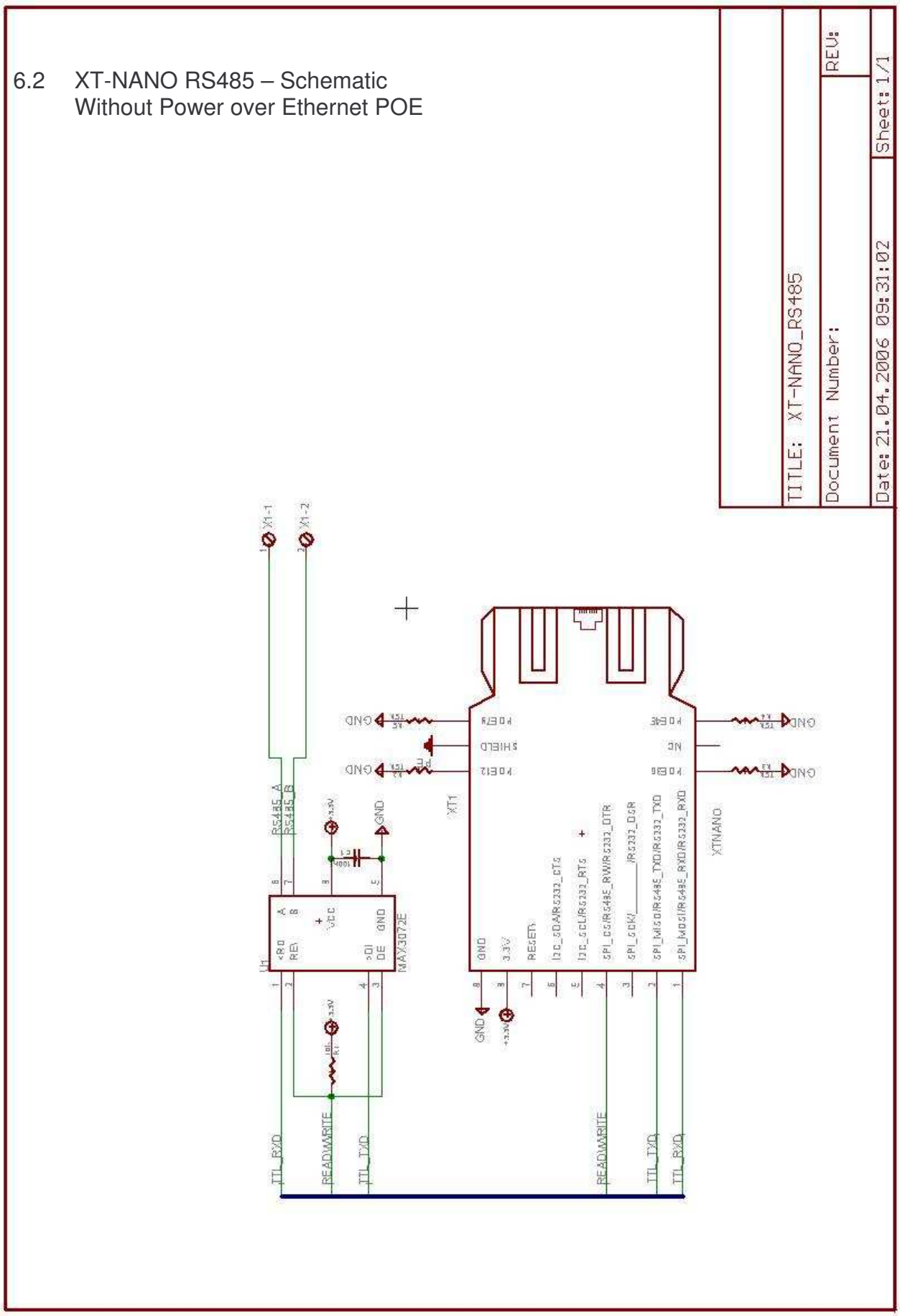
- Transmission speeds up to 12.5 MHz
- Character representation: 8 bit
- Flow control : switchable

6 Schematics

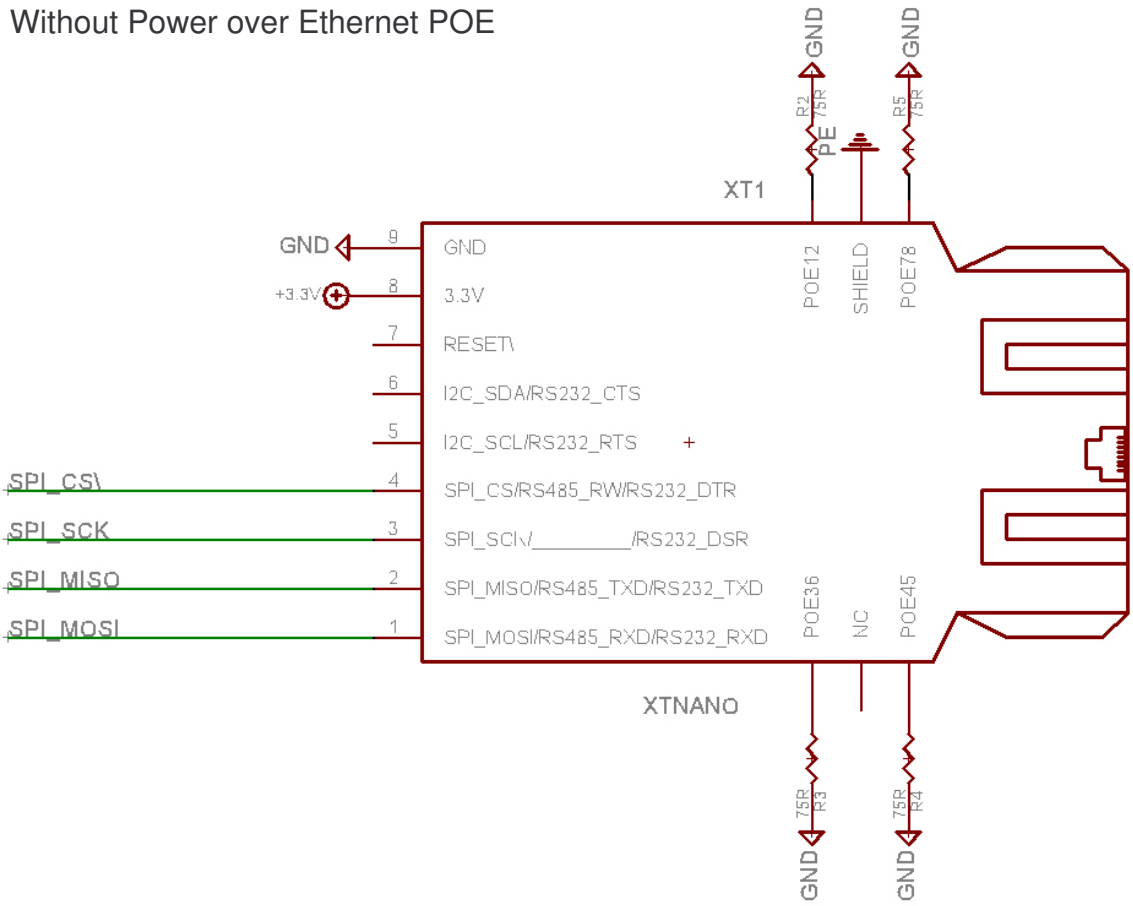
6.1 XT-NANO RS232 – Schematic Without Power over Ethernet POE



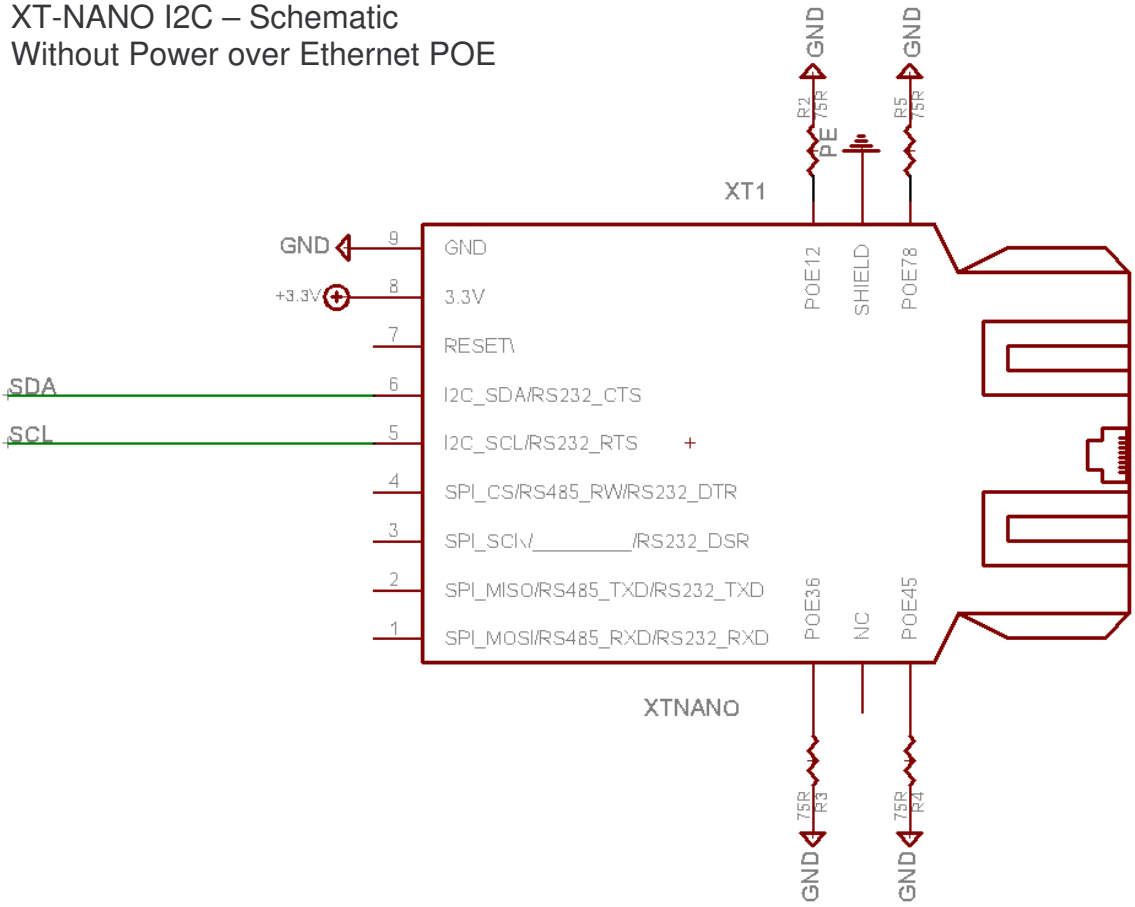
6.2 XT-NANO RS485 – Schematic Without Power over Ethernet POE



6.3 XT-NANO SPI – Schematic Without Power over Ethernet POE



6.4 XT-NANO I2C – Schematic Without Power over Ethernet POE



7 The I2C - BUS

Inter-Integrated Circuit (IIC)

The inter-IC bus (IIC) is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices. Being a two-wire device, the IIC bus minimizes the need for large numbers of connections between devices, and eliminates the need for an address decoder.

IIC Frequency Divider Register (IBFD)

Module Base + $1

	7	6	5	4	3	2	1	0
R W	IBC7	IBC6	IBC5	IBC4	IBC3	IBC2	IBC1	IBC0
RESET:	0	0	0	0	0	0	0	0

Figure 10-3. IIC Bus Frequency Divider Register (IBFD)

Read and write anytime

IBC7–IBC0 — I-Bus Clock Rate 7–0

This field is used to prescale the clock for bit rate selection. The bit clock generator is implemented as a prescale divider - IBC7-6, prescaled shift register - IBC5-3 select the prescaler divider and IBC2-0 select the shift register tap point. The IBC bits are decoded to give the tap and prescale values as shown in Table 10-2.

Table 10-2. I-Bus Tap and Prescale Values

IBC2-0 (bin)	SCL Tap (clocks)	SDA Tap (clocks)
000	5	1
001	6	1
010	7	2
011	8	2
100	9	3
101	10	3
110	12	4
111	15	4

IBC5-3 (bin)	scl2start (clocks)	scl2stop (clocks)	scl2tap (clocks)	tap2tap (clocks)
000	2	7	4	1
001	2	7	4	2
010	2	9	6	4
011	6	9	6	8
100	14	17	14	16
101	30	33	30	32
110	62	65	62	64
111	126	129	126	128

Table 10-3. Multiplier Factor

IBC7-6	MUL
00	01

Table 10-3. Multiplier Factor

IBC7-6	MUL
01	02
10	04
11	RESERVED

The number of clocks from the falling edge of SCL to the first tap (Tap[1]) is defined by the values shown in the scl2tap column of Table 10-2, all subsequent tap points are separated by 2^{IBC5-3} as shown in the tap2tap column in Table 10-2. The SCL Tap is used to generated the SCL period and the SDA Tap is used to determine the delay from the falling edge of SCL to SDA changing, the SDA hold time.

IBC7-6 defines the multiplier factor MUL. The values of MUL are shown in the Table 10-3.

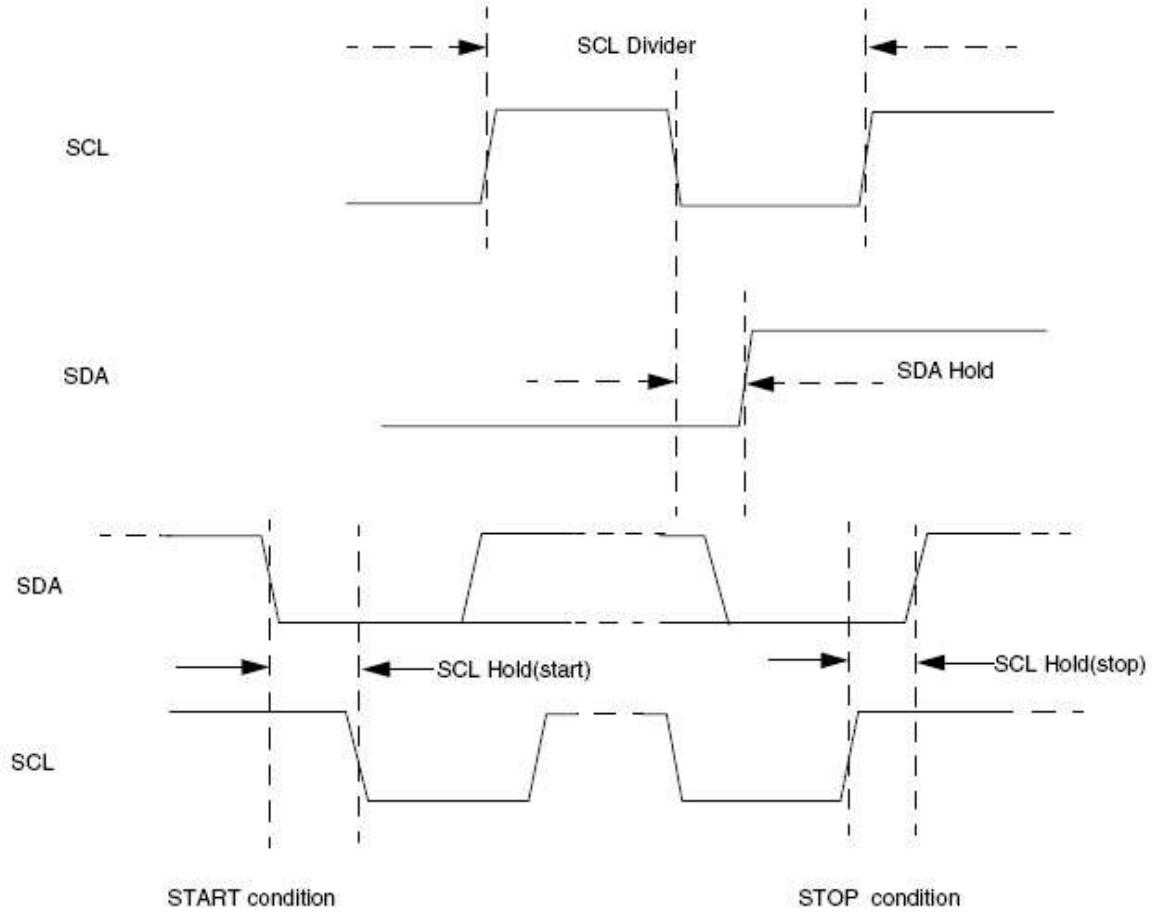


Figure 10-4. SCL Divider and SDA Hold

The equation used to generate the divider values from the IBFD bits is:

SCL Divider = MUL x {2 x (scl2tap + [(SCL_Tap -1) x tap2tap] + 2)}

The SDA hold delay is equal to the CPU clock period multiplied by the SDA Hold value shown in Table 10-4. The equation used to generate the SDA Hold value from the IBFD bits is:

SDA Hold = MUL x {scl2tap + [(SDA_Tap - 1) x tap2tap] + 3}

The equation for SCL Hold values to generate the start and stop conditions from the IBFD bits is:

SCL Hold(start) = MUL x [scl2start + (SCL_Tap - 1) x tap2tap]

SCL Hold(stop) = MUL x [scl2stop + (SCL_Tap - 1) x tap2tap]

Table 10-4. IIC Divider and Hold Values (Sheet 1 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=1				
00	20	7	6	11
01	22	7	7	12
02	24	8	8	13
03	26	8	9	14
04	28	9	10	15
05	30	9	11	16
06	34	10	13	18
07	40	10	16	21
08	28	7	10	15
09	32	7	12	17
0A	36	9	14	19
0B	40	9	16	21
0C	44	11	18	23
0D	48	11	20	25
0E	56	13	24	29
0F	68	13	30	35
10	48	9	18	25
11	56	9	22	29
12	64	13	26	33
13	72	13	30	37
14	80	17	34	41
15	88	17	38	45
16	104	21	46	53
17	128	21	58	65
18	80	9	38	41
19	96	9	46	49
1A	112	17	54	57
1B	128	17	62	65
1C	144	25	70	73

Table 10-4. IIC Divider and Hold Values (Sheet 2 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
1D	160	25	78	81
1E	192	33	94	97
1F	240	33	118	121
20	160	17	78	81
21	192	17	94	97
22	224	33	110	113
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2F	960	129	478	481
30	640	65	318	321
31	768	65	382	385
32	896	129	446	449
33	1024	129	510	513
34	1152	193	574	577
35	1280	193	638	641
36	1536	257	766	769
37	1920	257	958	961
38	1280	129	638	641
39	1536	129	766	769
3A	1792	257	894	897
3B	2048	257	1022	1025
3C	2304	385	1150	1153
3D	2560	385	1278	1281
3E	3072	513	1534	1537
3F	3840	513	1918	1921
MUL=2				
40	40	14	12	22
41	44	14	14	24
42	48	16	16	26
43	52	16	18	28
44	56	18	20	30
45	60	18	22	32

Table 10-4. IIC Divider and Hold Values (Sheet 3 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
46	68	20	26	36
47	80	20	32	42
48	56	14	20	30
49	64	14	24	34
4A	72	18	28	38
4B	80	18	32	42
4C	88	22	36	46
4D	96	22	40	50
4E	112	26	48	58
4F	136	26	60	70
50	96	18	36	50
51	112	18	44	58
52	128	26	52	66
53	144	26	60	74
54	160	34	68	82
55	176	34	76	90
56	208	42	92	106
57	256	42	116	130
58	160	18	76	82
59	192	18	92	98
5A	224	34	108	114
5B	256	34	124	130
5C	288	50	140	146
5D	320	50	156	162
5E	384	66	188	194
5F	480	66	236	242
60	320	34	156	162
61	384	34	188	194
62	448	66	220	226
63	512	66	252	258
64	576	98	284	290
65	640	98	316	322
66	768	130	380	386
67	960	130	476	482
68	640	66	316	322
69	768	66	380	386
6A	896	130	444	450
6B	1024	130	508	514
6C	1152	194	572	578
6D	1280	194	636	642
6E	1536	258	764	770
6F	1920	258	956	962

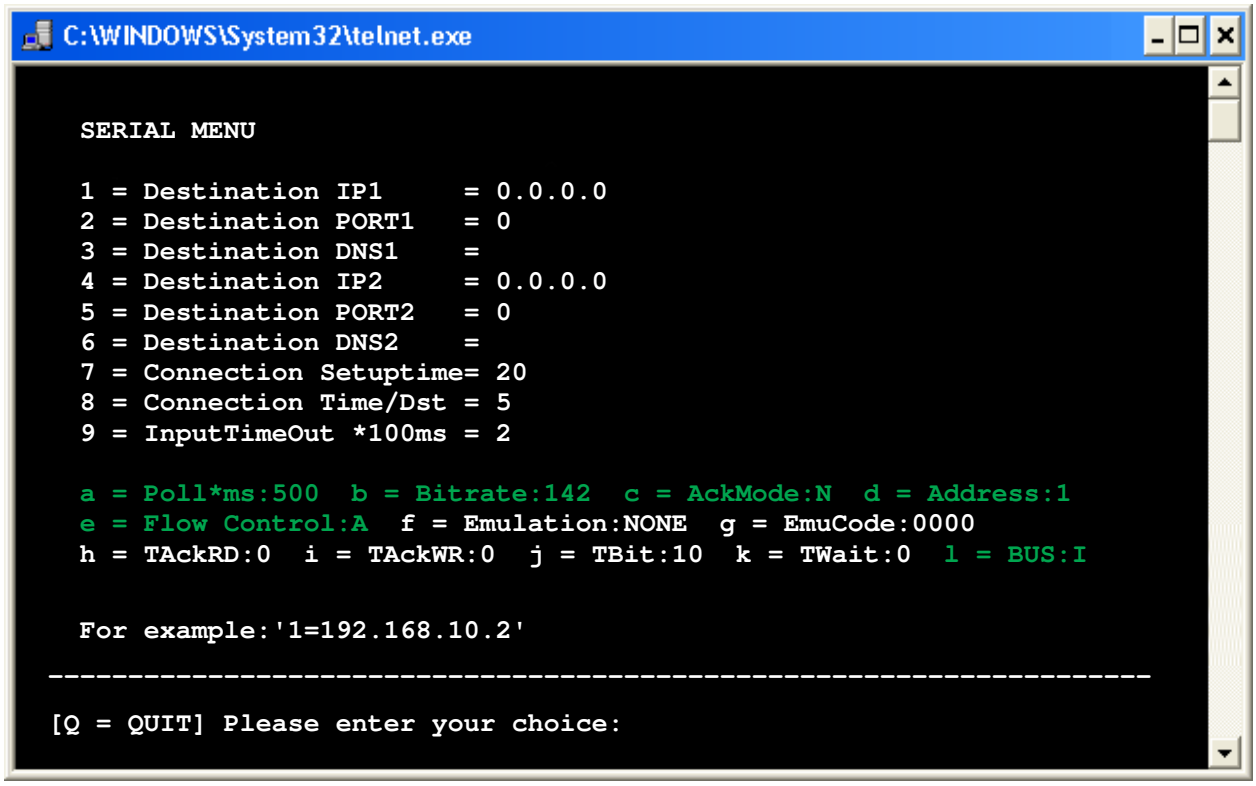
Table 10-4. IIC Divider and Hold Values (Sheet 4 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
70	1280	130	636	642
71	1536	130	764	770
72	1792	258	892	898
73	2048	258	1020	1026
74	2304	386	1148	1154
75	2560	386	1276	1282
76	3072	514	1532	1538
77	3840	514	1916	1922
78	2560	258	1276	1282
79	3072	258	1532	1538
7A	3584	514	1788	1794
7B	4096	514	2044	2050
7C	4608	770	2300	2306
7D	5120	770	2556	2562
7E	6144	1026	3068	3074
7F	7680	1026	3836	3842
MUL=4				
80	80	28	24	44
81	88	28	28	48
82	96	32	32	52
83	104	32	36	56
84	112	36	40	60
85	120	36	44	64
86	136	40	52	72
87	160	40	64	84
88	112	28	40	60
89	128	28	48	68
8A	144	36	56	76
8B	160	36	64	84
8C	176	44	72	92
8D	192	44	80	100
8E	224	52	96	116
8F	272	52	120	140
90	192	36	72	100
91	224	36	88	116
92	256	52	104	132
93	288	52	120	148
94	320	68	136	164
95	352	68	152	180
96	416	84	184	212
97	512	84	232	260
98	320	36	152	164

Table 10-4. IIC Divider and Hold Values (Sheet 5 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
99	384	36	184	196
9A	448	68	216	228
9B	512	68	248	260
9C	576	100	280	292
9D	640	100	312	324
9E	768	132	376	388
9F	960	132	472	484
A0	640	68	312	324
A1	768	68	376	388
A2	896	132	440	452
A3	1024	132	504	516
A4	1152	196	568	580
A5	1280	196	632	644
A6	1536	260	760	772
A7	1920	260	952	964
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
B0	2560	260	1272	1284
B1	3072	260	1528	1540
B2	3584	516	1784	1796
B3	4096	516	2040	2052
B4	4608	772	2296	2308
B5	5120	772	2552	2564
B6	6144	1028	3064	3076
B7	7680	1028	3832	3844
B8	5120	516	2552	2564
B9	6144	516	3064	3076
BA	7168	1028	3576	3588
BB	8192	1028	4088	4100
BC	9216	1540	4600	4612
BD	10240	1540	5112	5124
BE	12288	2052	6136	6148
BF	15360	2052	7672	7684

SETUP I2C



L = BUS: I
Use I2C – BUS

C=AckMode:N
N=Read acknowledge but don't evaluate it.
Y= Read acknowledge and evaluate it.

B=BitRate

The value **Bitrate** is the **I2C Frequency Divider Register**
The standard value is 142 = 8E = 100KHz
Note: The frequency of the XT-NANO = 25MHz

A=Poll * ms: 500
E=FlowControl:A

FlowControl:
A = with Protocol
B = without Protocol

Protocol:
In this case a query is made all **Poll * ms**. The interface send the value 80 to the slave address and expect the value 0. If the interface get back no answer or a value < > 0, the interface is not ready and a connection via TCP/IP can not occur. If the interface receives the Value 0 then the interface sends a Read – Command to the salve.
The read instruction has following format: The interface sends the value 83 to the slave address and expects the two Values (Read-Command) xx yy from the slave. The Values XX and YY are the High-Byte(XX) and LOW-Byte(YY) of Data the interface can read from the slave. Is the Value between 1 and 1024 (0x00,0x01 .. 0x04,0x00) then the interface read immediately Data from the Slave without sending a new Address(Read-Command).
If the interface receives Data via TCP/IP we send a Write-Command in following format: The interface sends the value 84 XX YY to the slave address and send immediately Data. The Values XX and YY are the High-Byte(XX) and LOW-Byte(YY) of Data will send to slave.

NOTE:
If the slave does not generate an acknowledge, we interrupt every command.

Polling:
→ **0x02** = command write Slave1 (I2C – Protocol)
→ 0x80
→ **0x03** = command read Slave1 (I2C – Protocol)
← 0x00 = ready
← 0x01 – 0xFF or no acknowledge = not ready

Read 1 Byte:
→ **0x02** = command write Slave1 (I2C – Protocol)
→ 0x83
→ **0x03** = command read Slave1 (I2C – Protocol
← 0x00,0x01
← 0x41 = read “A”

Send 2 Byte:
→ **0x02** = command write Slave1 (I2C – Protocol)
→ 0x84 0x00,0x02
→ 0x41,0x42 = send “AB”

without Protocol:
In this case a query is made all **Poll * ms**. The interface send a read command to the slave address and expect one Data. If the interface get back no acknowledge, we interrupt the command.
If the interface receives Data via TCP/IP the interface send a write command to the slave address and send immediately the count of Data.

NOTE:
If the slave does not generate an acknowledge, we interrupt every command.

Read 1 Byte:
→ **0x03** = command read Slave1 (I2C – Protocol)
← 0x41 = read “A”

Send n Byte:
→ **0x02** = command write Slave1 (I2C – Protocol
→ 0x41,0x42 = send “AB”

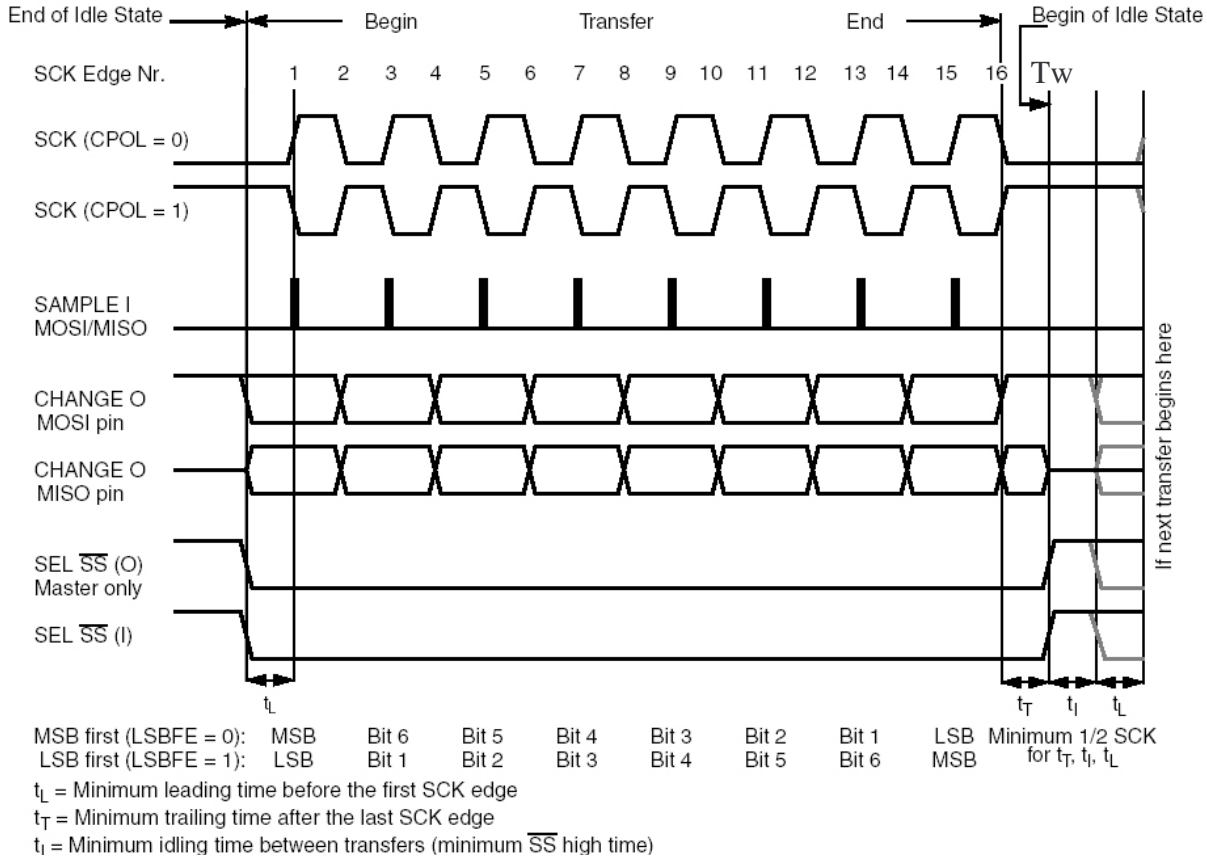
NOTE: if you set the Polling to “0”, we send no read commands. (Polling disable)

8 The SPI - BUS

Serial Peripheral Interface (SPI)

The SPI-bus is a 4-wire serial communications interface used by many microprocessor peripheral chips. The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that is standard across many microprocessors and other peripheral chips. It provides support for a bandwidth (**12,5 MHz**) network connection amongst CPUs and other devices supporting the SPI.

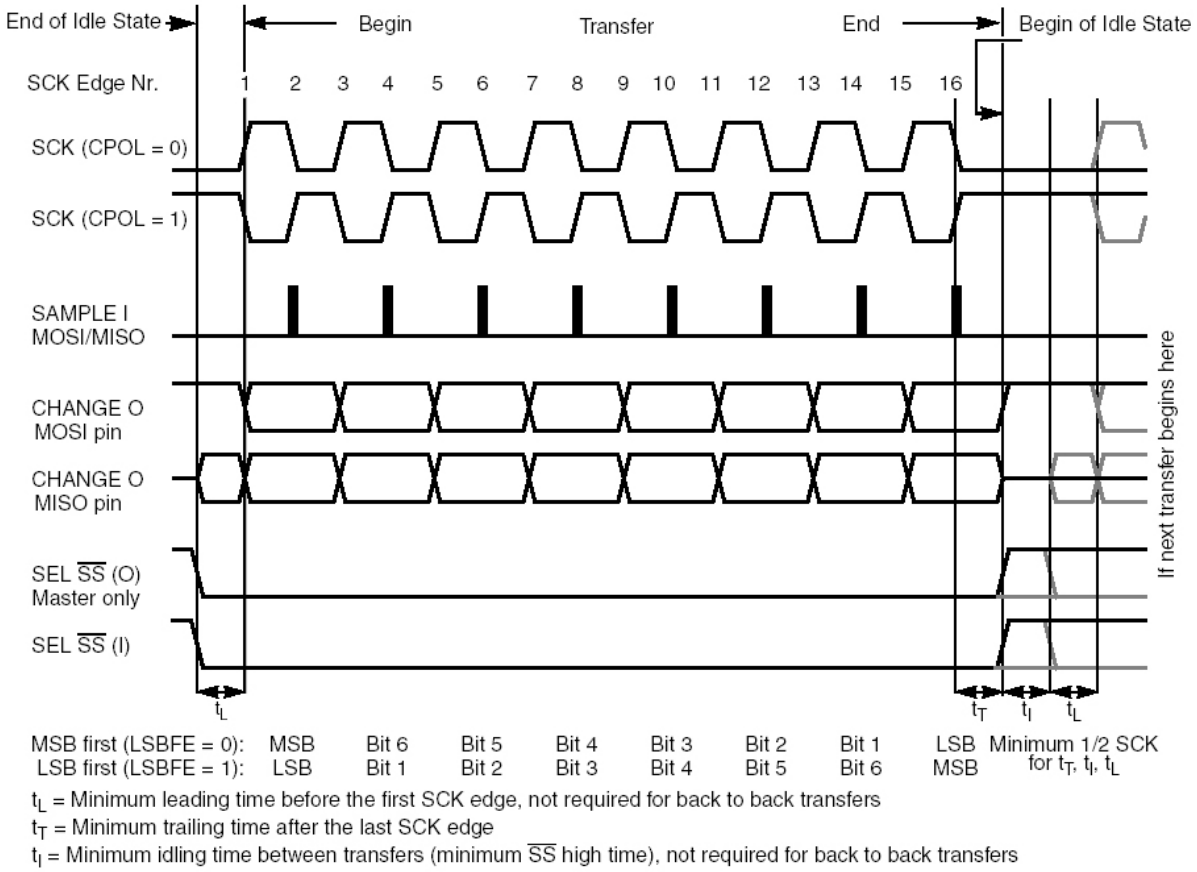
SPI bus is basically a relatively simple synchronous serial interface for connecting low speed external devices using quite minimal number of wires. SPI (serial peripheral interface) is an interface standard defined by Motorola. A synchronous clock shifts serial data into and out of the microcontrollers in blocks of 8 bits.



Timing:

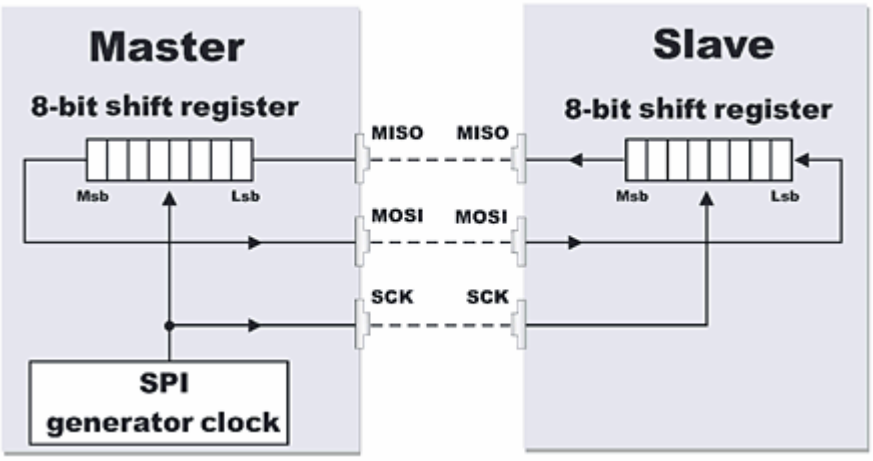
Tw = Timing Wait = 1us + Tb x 0,4us

CPHA = 1 Format

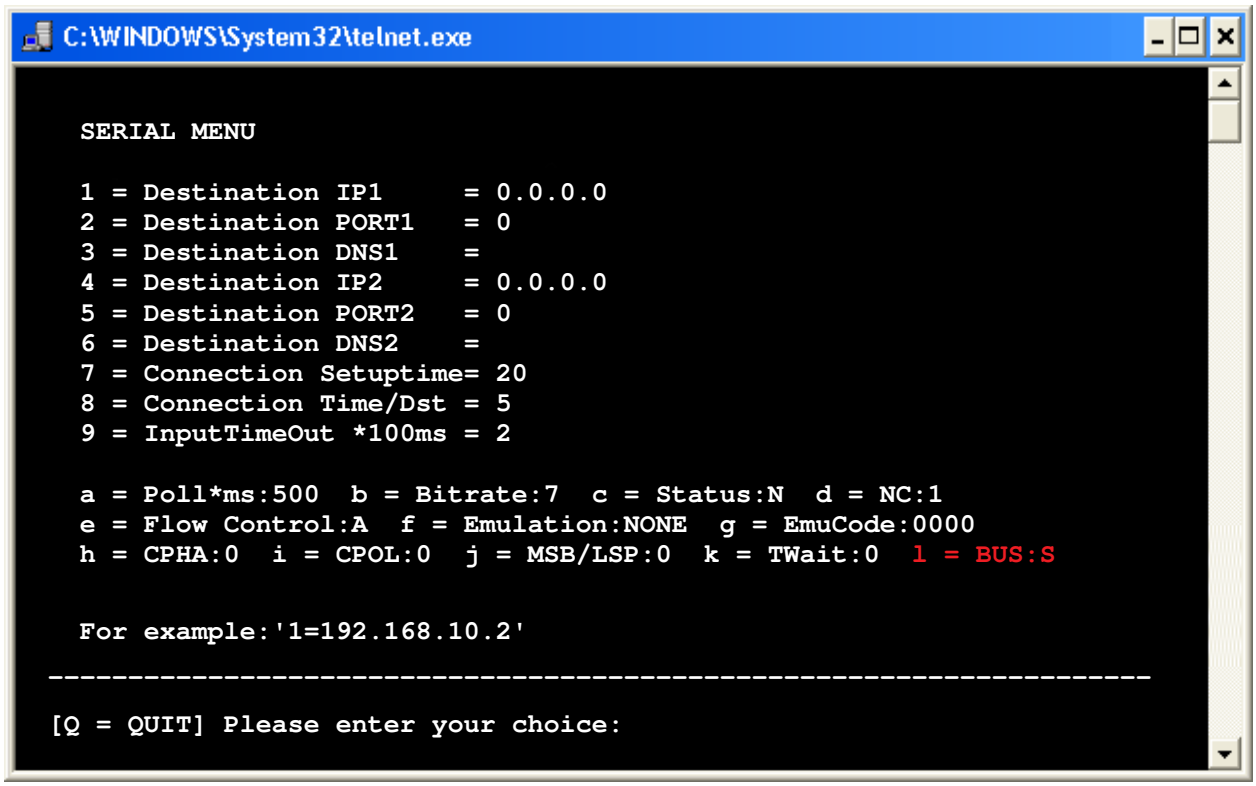


t_L = Minimum leading time before the first SCK edge, not required for back to back transfers
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back to back transfers

The procedure:



SETUP SPI (only XT-NANO)



L = BUS: S
Use SPI – BUS

B=Bitrate:

0	12.5 MHz
1	6.25 MHz
2	3.125 MHz
3	1.5625 MHz
4	781.25 kHz
5	390.63 kHz
6	195.31 kHz
7	**97.66 kHz**
112	1.5625 MHz
113	781.25 kHz
114	390.63 kHz
115	195.31 kHz
116	97.66 kHz
117	48.83 kHz
118	24.41 kHz
119	12.21 kHz

D=NC Not used

I=CPOL — SPI Clock Polarity Bit
This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values.
1 = Active-low clocks selected.
0 = Active-high clocks selected.

H=CPHA — SPI Clock Phase Bit
This bit is used to select the SPI clock format.
1 = Sampling of data occurs at even edges (2,4,6,...,16) of the SCK clock
0 = Sampling of data occurs at odd edges (1,3,5,...,15) of the SCK clock

J=MSB/LSB — LSB-First Enable
This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7.
1 = Data is transferred least significant bit first.
0 = Data is transferred most significant bit first.

C=Status:N
N = Read no status
Y = Read status

A=Poll * ms:500
E=FlowControl:A

FlowControl:

A	**=**	with Protocol
B	**=**	without Protocol and send back **every** received byte.
C	**=**	with Protocol
D	**=**	without Protocol and send back **no** received byte.

With Protocol:
In this case a query is made all **Poll * ms**. If **Status =Y,** the interface WriteRead a 0x80 to the slave address and expect the value 0x80 by the next WriteRead 0x00. Then the interface WriteRead 0x00 and will get back the status. If the status is 0x00 the interface knows slave is ready. If the value < > 0, the interface is not ready and a connection via TCP/IP can not occur.
If the status is ready, the interface sends a Read – Command to the salve. The read instruction has following format: We WriteRead a 0x83 to the slave and expect a 0x83 and the two Values xx yy from the slave by the next three WriteRead 0x00.The Values XX and YY are the High-Byte(XX) and LOW-Byte(YY) of Data the interface can read from the slave. Is the Value between 1 and 1024 (0x00,0x01 .. 0x04,0x00) then the interface read immediately the count of Data from the Slave by sending permanent WriteRead 0x00. If the interface receive Data via TCP/IP the interface send a Write-Command in following format: The interface send the value 0x84, 0x00, XX,YY, 0x00 to the slave and expect during that a 0xnn, 0x84, 0xnn, 0xnn, 0x84. Then the interface sends immediately the Data.

NOTE:
If the slave does not send the correct receipt, we interrupt every command.

Polling: (**Status =Y)**
→ 0x80 0x00 0x00
← 0xnn 0x80 0x00 = ready
← 0xnn 0x80 0x01 = not ready

Read 1 Byte:
→ 0x83 0x00 0x00 0x00 0x83
← 0xnn 0x83 0x00 0x01 0xnn
→ 0x00
← 0x41 = read “A”

Read n Byte:
→ 0x83 0x00 0x00 0x00 0x83
← 0xnn 0x83 0xHH 0xLL 0xnn
→ 0x00 0x00 ….
← 0x41 0x42 …. = read “AB….”

Send n Byte:
→ 0x84 0x00 0xHH 0xLL 0x00
← 0xnn 0x84 0xnn 0xnn 0x84
→ 0x41 0x42 ….
← 0xnn 0xnn …. = send “AB….”

without Protocol:
In this case a query is made all **Poll * ms**. The interface WriteRead a 0x00 to the slave and expect one Data. If the **FlowControl** is **D** then no received byte process or returned via TCP/IP. If the interface receives Data via TCP/IP then the interface WriteRead every byte directly to the slave. If the **FlowControl** is **D** than no received byte process or returned via TCP/IP.

Read 1 Byte:
→ **0x00**
← 0x41 = read “A”

Send n Byte:
→ **0x41 0x41**
← 0xnn 0xnn = send “AB”

NOTE: if you set the Polling to “0”, we send no read commands. (Polling disable).

9 The TTL IO

The TTL input/output mode provides total 6 I/O lines for input /output TTL digital signal. Each I/O pin can be set either input or output by user's program.

Features:
contains 6 I/O lines
Input/output selectable for each pin.
Standard +3.3V TTL digital signal.

Control the IO – Pins of the XT-NANO:
You must send always two bytes. The first is the command byte and the second byte is the pin value

For example:
In order to configure pin RXD to an output, you must send following control characters:

0x04 0x01 = Configure pin to Output
0x02 0x01 = set pin
0x01 0x01 = clear pin

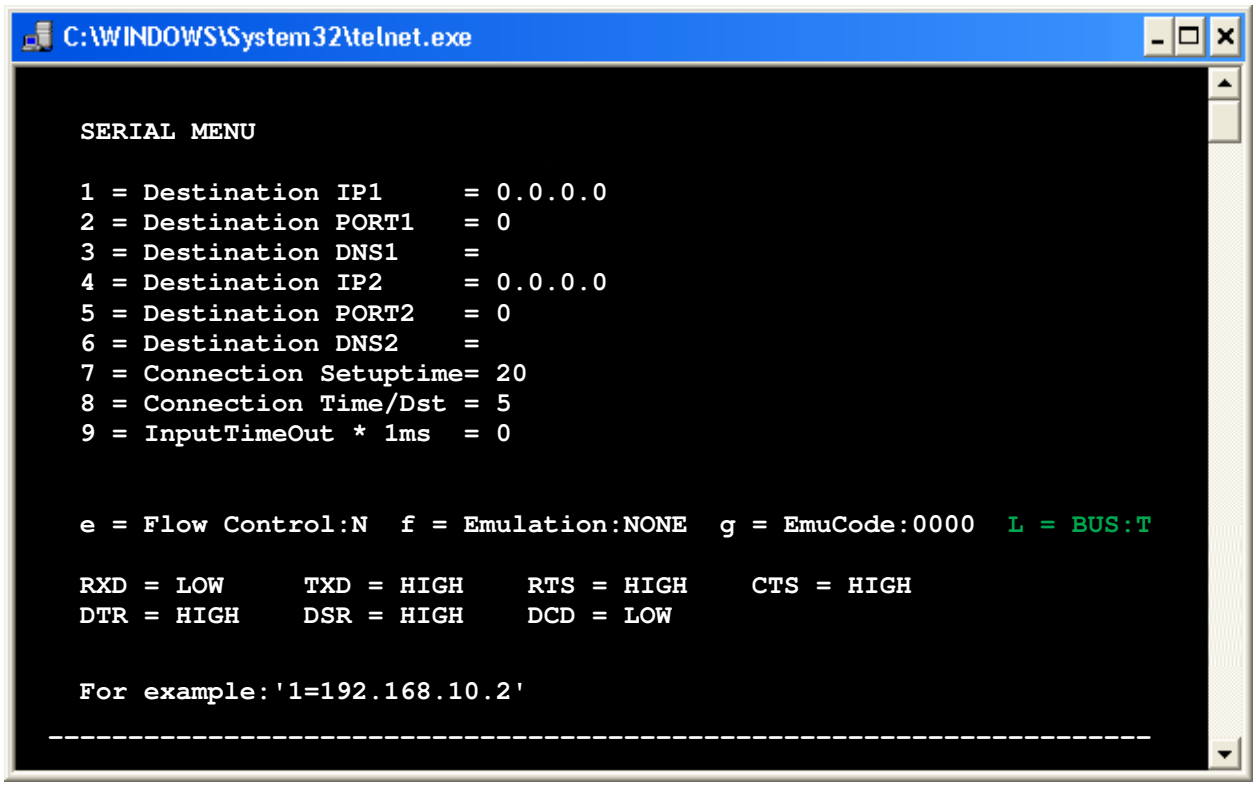
Commands:

0x00 = Read Pins
0x01 = Clear Pins
0x02 = Set Pins
0x03 = Configure Pin to Input
0x04 = Configure pin to Output
0x05 = Clear Pullup
0x06 = Set Pullup

Pin Value

BIT 0 = Pin RXD
BIT 1 = Pin TXD
BIT 2 = Pin RTS
BIT 3 = Pin CTS
BIT 4 = Pin DTR
BIT 5 = Pin DSR
BIT 6 = Not in use
BIT 7 = Not in use

SETUP TTL IO



(RXD to DCD shows the state)

10 POE Power over Ethernet

For POE (Power over Ethernet) we prepared everything. If you want to use POE (Power over Ethernet), you find out then here:

http://poweroverethernet.com