

# **QBridge**

## **I2C, SPI, CAN Control Software User's Manual**

**Date: 5-5-2005  
Rev 1.1**

<b>1</b>	<b><i>Introduction</i></b> .....	<b>1</b>
1.1	What QBridge can do? .....	1
1.2	Disclaimer .....	1
1.3	Operational Format .....	1
<b>2</b>	<b><i>How to use QBridge</i></b> .....	<b>1</b>
2.1	Connect to Hyper terminal .....	1
2.2	Power up the QBridge .....	2
2.3	Pins layout .....	2
2.4	Diagram to connect to SPI or I2C device.....	3
2.5	Diagram to connect to CAN device .....	3
<b>3</b>	<b><i>Main Menu</i></b> .....	<b>4</b>
3.1	Command I (if I2C feature is purchased).....	4
3.2	Command C (if CAN feature is purchased) .....	4
3.3	Command S (if SPI feature is purchased) .....	4
3.4	Command B .....	4
3.5	Reset Status.....	5
<b>4</b>	<b><i>I2C Application Menu</i></b> .....	<b>6</b>
4.1	Command S .....	6
4.2	Command P .....	6
4.3	Command T.....	6
4.4	Command B .....	6
4.5	Command W .....	7
4.6	Command G.....	7
4.7	Command Q.....	7
4.8	Command R .....	7
4.9	Command D .....	8
4.10	Command I.....	8
4.11	Command L.....	8
4.12	Command M.....	8
4.13	Command '?'.....	8
4.14	Error code listing.....	8
<b>5</b>	<b><i>CAN (controller area network) Application Menu</i></b> .....	<b>9</b>
5.1	Command C .....	9
5.2	Command W .....	9

5.3	Command B .....	10
5.4	Command O .....	10
5.5	Command T.....	11
5.6	Command D .....	11
5.7	Command S .....	11
5.8	Command A .....	12
5.9	Command M.....	12
5.10	Command N .....	12
5.11	Command L.....	12
5.12	Command '?'.....	12
5.13	CAN message receiving .....	12
<b>6</b>	<b><i>SPI application menu .....</i></b>	<b>13</b>
6.1	Command P .....	13
6.2	Command B .....	13
6.3	Command G .....	14
6.4	Command W .....	14
6.5	Command N .....	14
6.6	Command S .....	14
6.7	Command A .....	15
6.8	Command D .....	15
6.9	Command E .....	15
6.10	Command Q.....	15
6.11	Command M.....	16
6.12	Command L.....	16
6.13	Command '?'.....	16
6.14	Command X .....	16
<b>7</b>	<b><i>Revision History .....</i></b>	<b>16</b>

# 1 Introduction

## 1.1 What QBridge can do?

QBridge is a single board computer provides an efficient and economical way to connect I2C, SPI or CAN (optional feature) to the PC RS232 port. It quickly turns any PC into an I2C, SPI mater or CAN node.

## 1.2 Disclaimer

While every effort has been made to ensure that the information contained in this manual is accurate and complete, no liability can be accepted for any error and omission. Q-Proto system reserves the right to change the specification of the hardware and software described herein at any time prior notice.

Q-Proto System shall not be liable for direct, indirect, incidental, general or consequential damage from the use of the products from Q-Proto system. If you do not agree with these terms, do not buy the products.

## 1.3 Operational Format

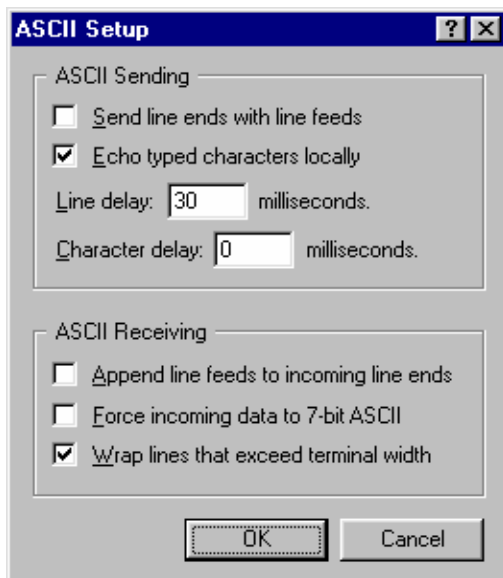
All the QBridge operation is based on HEX format.

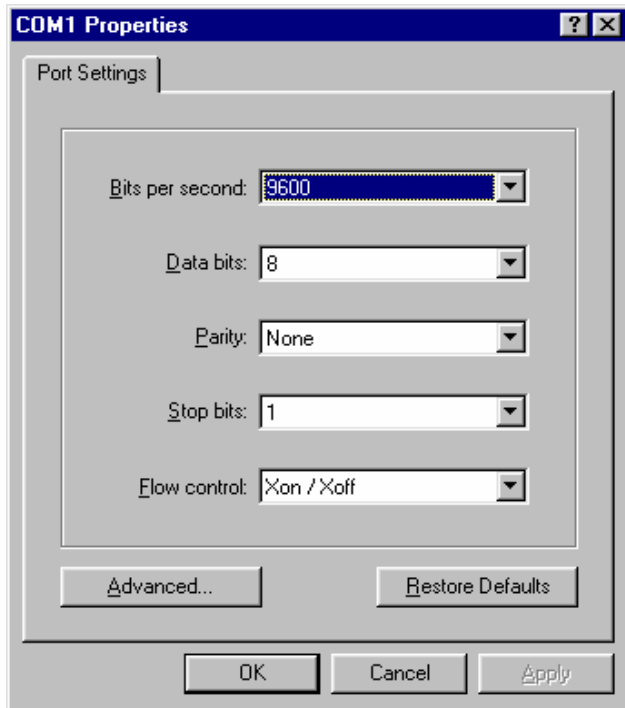
# 2 How to use QBridge

## 2.1 Connect to Hyper terminal

QBridge connects to PC via RS232 port. It uses Windows Hyper Terminal program to send commands and receive data. The setting for Hyper Terminal is 9600-baud rate, 8 data bits, 1 stop bit, no parity bit, Xon/Xoff flow control, and also check 'echo typed character locally' and set '30 milliseconds' to line delay.

Example setup is shown below:





## 2.2 Power up the QBridge

An external +9 V shall be applied to QBridge for its normal operation.

## 2.3 Pins layout

QBridge purchased after April 2005 has pins layout slightly different from previous version.

- Pins layout for QBridge purchased before April 2005

A female DB9 is for Host RS232 connection and a 9-pins header for external target.



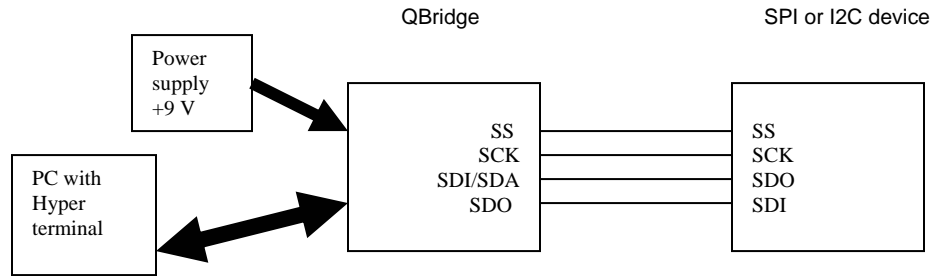
- Pins layout for QBridge purchased after May 2005



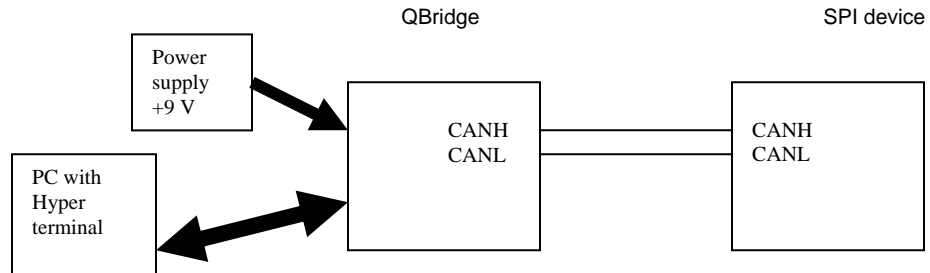
A female DB9 is for Host RS232 connection and a male DB9 connector is used to connect external target. Pins layout for external target (male DB9) is shown below:

5: +9V	9: Boot
4: GND	8:SS
3:SCK	7: SDI/SDA
4:SDO	6:CANH
1: CANL	

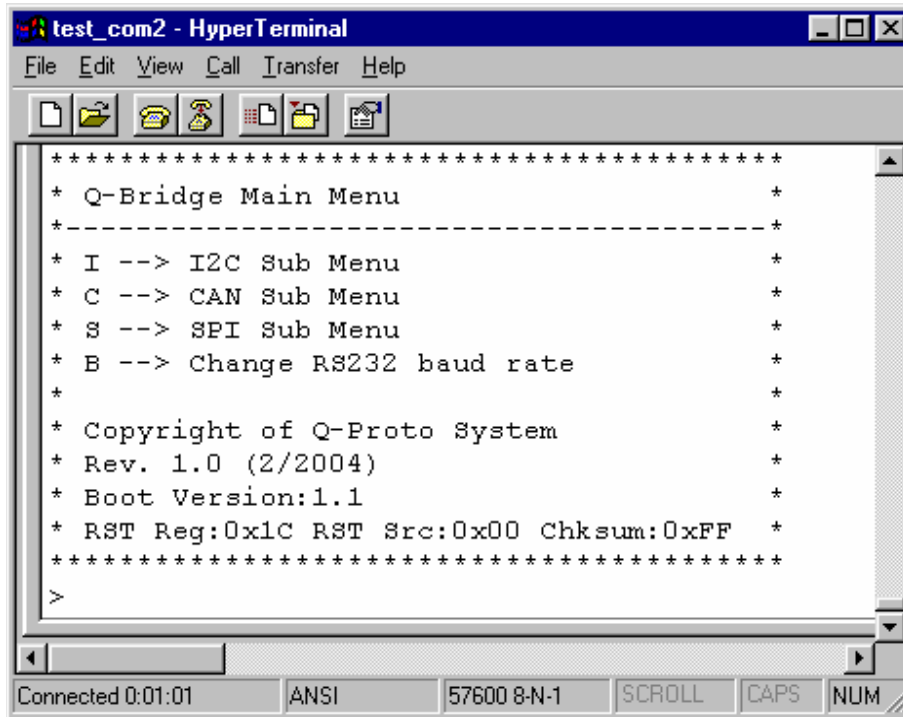
## 2.4 Diagram to connect to SPI or I2C device



## 2.5 Diagram to connect to CAN device



## 3 Main Menu



```
test_com2 - HyperTerminal
File Edit View Call Transfer Help
*****
* Q-Bridge Main Menu *
* ----- *
* I --> I2C Sub Menu *
* C --> CAN Sub Menu *
* s --> SPI Sub Menu *
* B --> Change RS232 baud rate *
* *
* Copyright of Q-Proto System *
* Rev. 1.0 (2/2004) *
* Boot Version:1.1 *
* RST Reg:0x1C RST Src:0x00 Chksum:0xFF *
*****
^
Connected 0:01:01 ANSI 57600 8-N-1 SCROLL CAPS NUM
```

### 3.1 Command I (if I2C feature is purchased)

I command will transfer the main menu to I2C sub menu.

### 3.2 Command C (if CAN feature is purchased)

C command will transfer the main menu to CAN (controller area network) sub menu.

### 3.3 Command S (if SPI feature is purchased)

S command will transfer the main menu to SPI sub menu.

### 3.4 Command B

B command will configure QBridge to different Baud rate.

- B0  
Set Baud rate to 9600 bps.
- B1  
Set Baud rate to 19200 bps.
- B2  
Set Baud rate to 57600 bps. This is default setting.
- B3  
Set Baud rate to 125K bps.

**Note:**

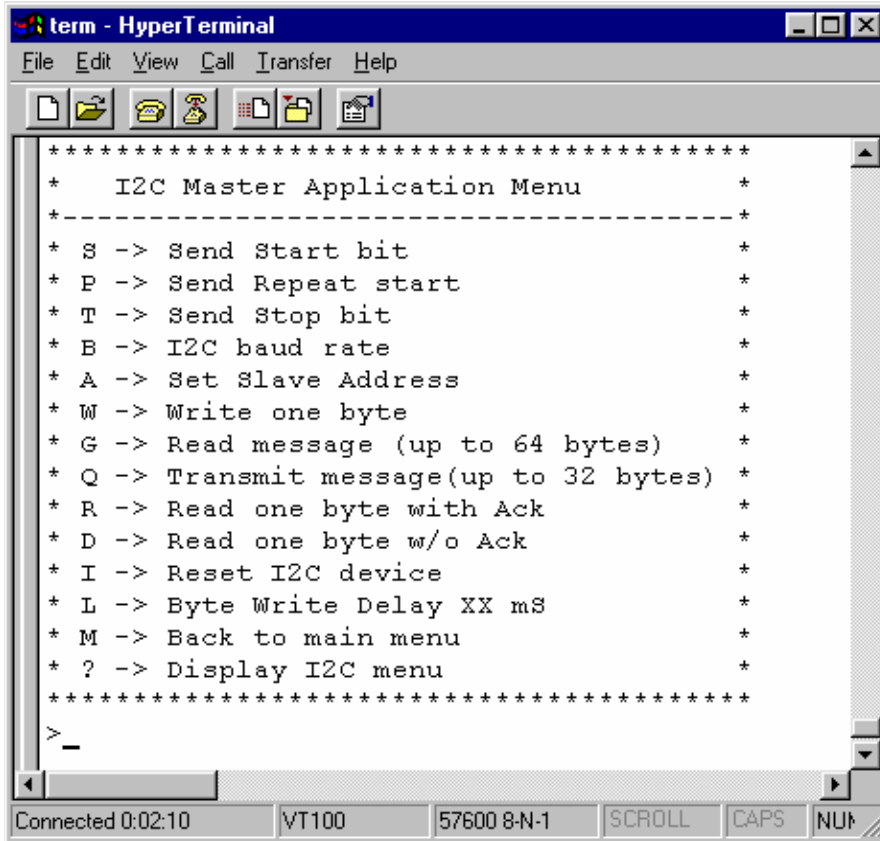
Power must be cycled in order for the new Baud rate to take effect.

### **3.5 Reset Status**

At main menu, it will show the current status of the reset register value, reset source and the software checksum. The reset register and reset source values are only for debug purpose when report problem to Q-proto system.



## 4 I2C Application Menu



```
term - HyperTerminal
File Edit View Call Transfer Help
*****
*   I2C Master Application Menu   *
*-----*
* S -> Send Start bit            *
* P -> Send Repeat start         *
* T -> Send Stop bit            *
* B -> I2C baud rate             *
* A -> Set Slave Address         *
* W -> Write one byte            *
* G -> Read message (up to 64 bytes) *
* Q -> Transmit message(up to 32 bytes) *
* R -> Read one byte with Ack    *
* D -> Read one byte w/o Ack    *
* I -> Reset I2C device         *
* L -> Byte Write Delay XX mS   *
* M -> Back to main menu        *
* ? -> Display I2C menu         *
*****
>
_
```

Connected 0:02:10 VT100 57600 8-N-1 SCROLL CAPS NUM

### 4.1 Command S

Command S will send the I2C start bit. If Start bit is not sent successfully, failure code can be found in Section 4.14. User shall ensure that STOP bit is sent first before the START bit is sent.

### 4.2 Command P

Command P will send the I2C repeated start bit. If the repeated Start bit is not sent successfully, failure code can be found in Section 4.14.

### 4.3 Command T

Command T will send the I2C Stop bit that terminates communication. If Stop bit is not sent successfully, failure code can be found in Section 4.14.

### 4.4 Command B

Command B will set the I2C clock speed



## 4.9 Command D

Command D reads one byte from slave without acknowledge bit at the end. User shall use command W to specify slave and word address if a particular location is accessed.

Format: D

## 4.10 Command I

Command I reset I2C controller to the known state.

I2C salve address will be set to 0x50 and delay between each write is 0xA ms.

I2C clock speed will be first fetched from NVRAM or set to 100K if NVRAM value is corrupted.

## 4.11 Command L

Command L will set the interval between each data byte sent to salve. Because I2C is used often in EEPROM application, a internal write delay is needed for EEPROM programming.

Format: L xx where xx is number of millisecond delay in hex format.

Example: L 10 Set write delay to 16 ms

## 4.12 Command M

Command M will transfer the current I2C sub menu to main menu.

Format: >M

## 4.13 Command '?'

Command '?' will display the entire I2C sub menu again.

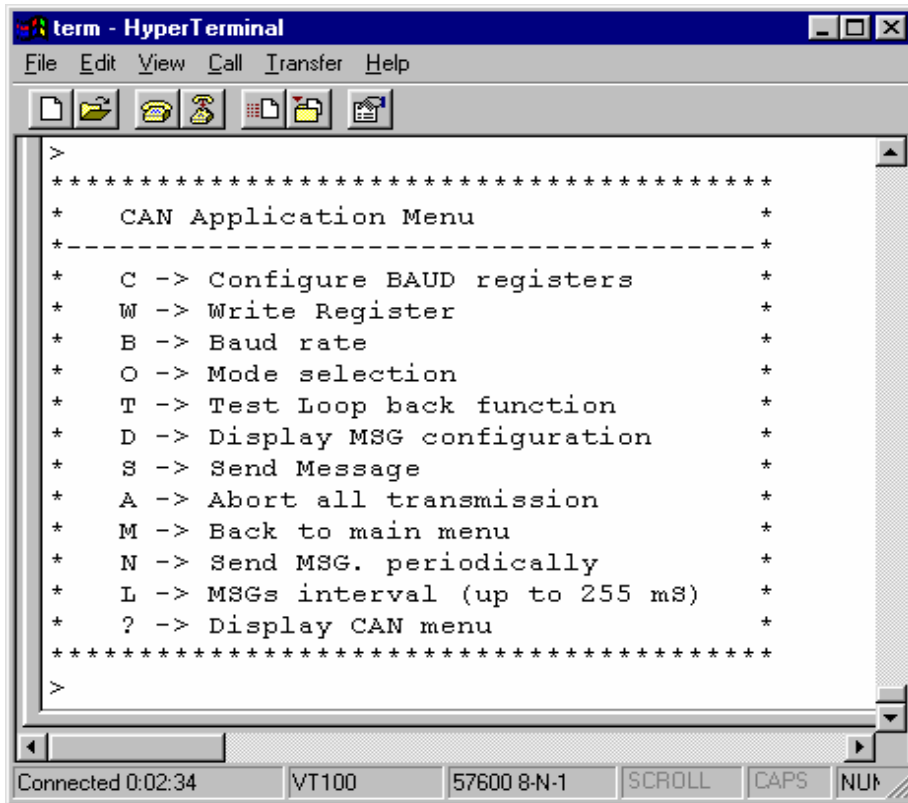
Format: >?

## 4.14 Error code listing

One of the following error codes will be returned if command is failed to execute.

Error code	Description
1	Time out when sending data or start bit
2	Timeout when sending STOP bit
4	No ACK from slave
8	Receive overflow
0x10	Timeout when reading data
0x20	Timeout when acking to slave (ACKDT)
0x40	Bus collision

## 5 CAN (controller area network) Application Menu



The screenshot shows a HyperTerminal window titled "term - HyperTerminal". The window contains a menu of commands for the CAN application. The menu is enclosed in a box with asterisks and dashes. The commands listed are:

```
>
*****
*   CAN Application Menu   *
*-----*
*   C -> Configure BAUD registers *
*   W -> Write Register      *
*   B -> Baud rate          *
*   O -> Mode selection     *
*   T -> Test Loop back function *
*   D -> Display MSG configuration *
*   S -> Send Message       *
*   A -> Abort all transmission *
*   M -> Back to main menu   *
*   N -> Send MSG. periodically *
*   L -> MSGs interval (up to 255 mS) *
*   ? -> Display CAN menu   *
*****
>
```

At the bottom of the window, there is a status bar with the following information: Connected 0:02:34, VT100, 57600 8-N-1, SCROLL, CAPS, NUM.

### 5.1 Command C

Command C will let you configure the Baud rate registers for PIC18F258. Please refer to the PIC18F258 datasheet for details.

Format:

- C aa,bb,cc,dd,ee

Where aa is Baud rate prescaler bits, bb is Synchronized jump width bit, cc is propagation time select bits, dd is phase segment 1 bit, ee is phase segment 2 bit.

### 5.2 Command W

Command W will allow you write value to CAN receive filter and mask registers. Note that QBridge CAN receiver module is always active. Once a matched CAN message is received, it will be shown on terminal.

Format:

- W r {m0, m1, f0, f1, f2}, nnnn,f{s,x,r,t}

Where r is register to write, nnnn is the value in hex format, f is the framing format bit.

Register r definition:

m0: receive buffer 0 mask register

m1: receive buffer 1 mask register  
f0: receive buffer 0 filter 0 register  
f1: receive buffer 0 filter 1 register  
f2: receive buffer 1 filter 2 register  
f3: receive buffer 1 filter 3 register  
f4: receive buffer 1 filter 4 register  
f5: receive buffer 1 filter 5 register

Framing bit definition:  
S: Standard frame  
X: Extended frame  
R: Remote standard frame  
T: Remote extended frame

Example: w m0, 34ef,x      Configure mask 0 register to extended frame with value 0x34e.

### 5.3 Command B

Command B will configure the baud rate. Power on default is 125K

Format:

- B

Display the current baud rate. Note that baud rate will not be shown properly if command C is used to set to customized baud rate prior.

- B 0

Set baud rate to 1M bits

- B 1

Set baud rate to 800 K bits

- B 2

Set baud rate to 500 K bits

- B 3

Set baud rate to 250 K bits

- B 4

Set baud rate to 125 K bits

- B 5

Set baud rate to 100 K bits

- B 6

Set baud rate to 50 K bits

### 5.4 Command O

Command O will change the CAN operation mode

Format: O n{0,1,2,3,4} Where n is the mode to change to

- O

Show the current CAN operation mode

- O 0

Set to configuration mode

- O 1

Set to disable mode

- O 2

Set to operation mode

- O 3

Set to listen mode

- O 4

Set to loop back mode

## 5.5 Command T

Command T is loop back test mode.

Format: >T

## 5.6 Command D

Command D will show CAN filter, mask, transmit and error register values.

Format: >D

Example:

```
>M0:X-0x1FFFFFFE S-0x000007FF F0:X-0x12345678 F1:S-0x00000678
>M1:X-0x1FFFFFFF S-0x000007FF
>F2:X-0x000001BC F3:S-0x000001BC F4:X-0x00000000 F5:X-0x00000000
>T0:S-0x00000678 FF EE DD 05 04 03 02 01
>T1:X-0x12345678 A5 34 31 88 77 99 EF 5A
>T2:XR-0x000001BC
>TxErr:00 RxErr:00 Invalid Msg. cntr:00 Bus Err. Cntr:00
>
```

Mask 0 is 0x1FFFFFFE for extended frame or 0x000007FF for standard frame

Filter 0 is extended frame 0x12345678

Filter 1 is standard frame 0x00000678

Mask 1 is 0x1FFFFFFE for extended frame or 0x000007FF for standard frame

Filter 2 is extended frame 0x000001BC

Filter 3 is extended frame 0x000001BC

Filter 4 and 5 are extended frame 0x00000000

Transmit buffer 0 is standard frame, id 0x00000678, data FF EE DD 05 04 03 02 01

Transmit buffer 1 is extended frame, id 0x12345678, data A5 34 31 88 77 99 EF 5A

Transmit buffer 3 is remote standard frame, id 0x000001BC

**Note:** CAN message framing format: X– standard frame, R- remote frame, S- standard frame.

## 5.7 Command S

Command S will transmit the CAN message to a specified buffer.

Format:

S x{0,1,2}, nnnn, f{x, s, r, t}, aa, bb, cc, dd

Where x is the buffer number, nnnn is the CAN id in hex format, f is the CAN message format bit, aa – dd are CAN data up to 8 bytes

Message format: {x, s, r, t} where x is extended frame, s is standard frame, r is remote standard frame, t is extended frame. Note that when remote frame is sent, user should still send dummy data byte even receiver ignores it, because some CAN application will expect remote frame data length is the same as returned data length.

Example:

S 0, 1234edf, x, 11,22, 3e, 4d

Sending extended frame 0x1234edf with data packet 11,22,3e,4d to transmit buffer 0.

S1, 3ef, t, 00,00,00

Sending extended remote frame, id: 0x3ef, data length is 3.

## 5.8 Command A

Command A aborts all current transmission actives.

Format: >A

## 5.9 Command M

Command M transfer from CAN application to Main menu.

Format: >m

## 5.10 Command N

Command N will continuously send a preset CAN message for a specified number of times. A delay is also inserted between messages sent and delay can be set with command L.

Format: N x, nn

Where x is the CAN transmit buffer number between 0 to 2, nn is number of times to repeat this message, which is up to 255 times.

Example: N 2, 20 send TX buffer 2 for 32 (hex 0x20) times.

## 5.11 Command L

Command L will set the interval between each sent message when messages are sent with command N. Power on default value is 0xA ms.

Format: > L 20 Set interval to 32 (0x20) milliseconds.

## 5.12 Command '?'

Command '?' will display the entire CAN application menu again.

## 5.13 CAN message receiving

Once CAN application is entered, CAN message receiver module is automatically enabled. Whenever a matched message is received, it will be displayed on the console. Message is shown in the following format:

<ID>-<Type>-<Filter> <Data0~Data8> <Timestamp> <Dropped packet counter>

- ID is the CAN message ID in hex format
- Type is the CAN message type. R is remote frame, X is extended frame, S is standard frame.
- Filter is the received hit filter ranges from 0-5
- Data0~Data8 is the CAN data packet
- Timestamp is the received CAN message timestamp in 2 microsecond resolution.

**Note:** The total of bits for standard CAN message is calculated as of following:

SOF+11bit ID+RTR+6 Control bits + number of data\*8 =16 bits CRC + 2 ACK + 7 EOF bits. Extended frame will have 29 bits ID instead of 11 bits.

- Dropped packet counter is the counter for all the dropped packets when received buffers are overrun.

### Example1:

>Recv:12345678-X-F00 A5 34 31 88 77 99 EF 5A T:131F D:00

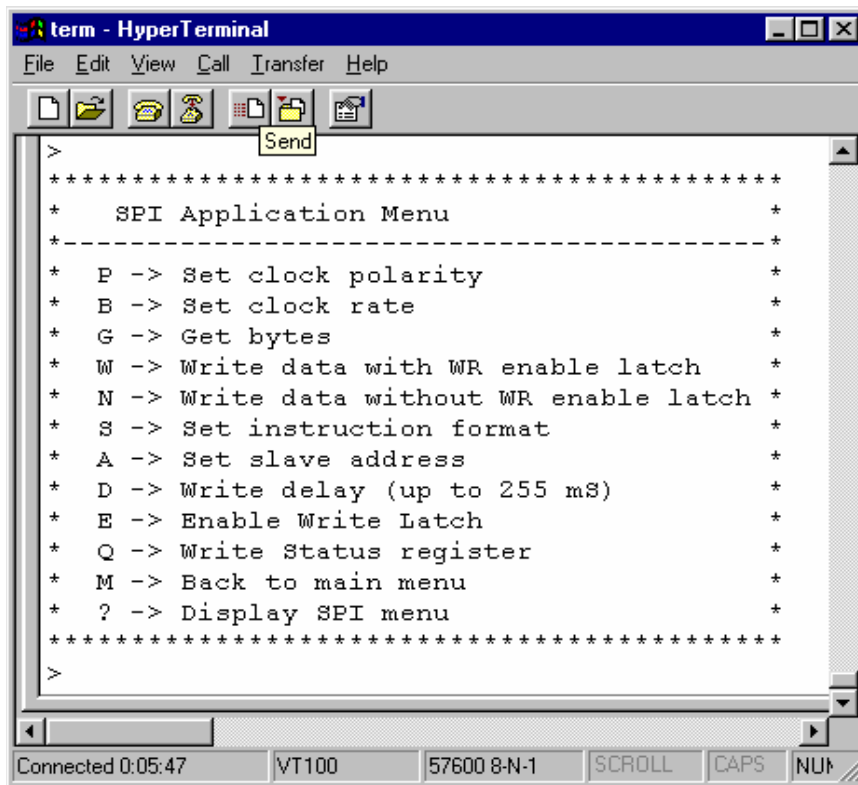
Received message: ID is 0x12345678, X is extended frame, from filter 0, packet is A5 34 31 88 77 99 EF 5A, timestamp is 0x131F, dropped packet counter is 0.

### Example2:

>Recv:000001BC-RX-F02 T:17A2 D:00

Received message: remote extended frame, ID is 0x1BC, timestamp is 0x17A2, no dropped packet.

## 6 SPI application menu



The screenshot shows a HyperTerminal window titled "term - HyperTerminal". The window contains a menu of SPI commands. The menu is enclosed in a box with asterisks and dashes. The commands listed are: P -> Set clock polarity, B -> Set clock rate, G -> Get bytes, W -> Write data with WR enable latch, N -> Write data without WR enable latch, S -> Set instruction format, A -> Set slave address, D -> Write delay (up to 255 mS), E -> Enable Write Latch, Q -> Write Status register, M -> Back to main menu, and ? -> Display SPI menu. The window also shows a status bar at the bottom with connection information: "Connected 0:05:47", "VT100", "57600 8-N-1", and buttons for "SCROLL", "CAPS", and "NUM".

```
>
*****
*   SPI Application Menu   *
*-----*
* P -> Set clock polarity *
* B -> Set clock rate    *
* G -> Get bytes         *
* W -> Write data with WR enable latch *
* N -> Write data without WR enable latch *
* S -> Set instruction format *
* A -> Set slave address  *
* D -> Write delay (up to 255 mS) *
* E -> Enable Write Latch *
* Q -> Write Status register *
* M -> Back to main menu  *
* ? -> Display SPI menu   *
*****
>
```

Connected 0:05:47 VT100 57600 8-N-1 SCROLL CAPS NUM

### 6.1 Command P

Command P sets SPI clock edge and polarity.

Format:

- P  
Shows the current SPI clock edge and polarity.
- P0  
SPI clock edge is idle low, data transmitted in falling edge.
- P1  
SPI clock edge is idle low, data transmitted in rising edge. This setting is power on default.
- P2  
SPI clock edge is idle high, data transmitted in rising edge.
- P3  
SPI clock edge is idle high, data transmitted in falling edge.

### 6.2 Command B

Command B sets the SPI clock speed rate.



Format:

- B  
Show the current speed. Power on default is 250 K

- B0  
Set clock speed to 4 MHz

- B1  
Set clock speed to 1 MHz

- B2  
Set clock speed to 250 KHz

### 6.3 Command G

Command G gets numbers of data from SPI slave device. Default address may be set using command A.

Format: > g 10                      gets 16 (0x10) bytes from device

### 6.4 Command W

Command W writes user input data to SPI slave device. This command will automatically enable the Write Enable Latch before each write to slave. An interval will be inserted between write cycles, which can be set with command D. Default address may be set with command A. Slave Address will be incremented after this operation.

Format: > w 12,34,56

If slave address is 0x0 before the operation, it will be set to 0x3 after the operation.

### 6.5 Command N

Command N is similar to command W except it won't issue the Enable Write Latch command between write cycles.

Format: > n 12,34,56

If slave address is 0x0 before the operation, it will be set to 0x3 after the operation.

### 6.6 Command S

Command S sets the control command format.

Format:

- S  
Shows the current control commands.

Power on default value:

Read command	0x3
Write command	0x2
Write disable command	0x4
Write enable command	0x6
Read status register command	0x5
Write status register command	0x1

Example:

```
>s  
>RD:34 WR:02 WRDI:04 WREN:06  
>RDSR:05 WRSR:01
```

RD is read command, WR is write command, WREN is write enable command, RDSR is read status register command and WRSR is write status register command.

- SR nn  
Sets the read command, where nn is the data in hex format.  
Example: > SR 34 Sets read command to 0x34.

- SW nn  
Sets the write command, where nn is the data in hex format
- SS nn  
Sets the read status register command, where nn is the data in hex format.
- ST nn  
Sets the write status register command, where nn is the data in hex format.
- SE nn  
Sets the enable write command, where nn is the data in hex format.
- SD nn  
Sets the disable write command, where nn is the data in hex format.

## 6.7 Command A

Command A sets the SPI slave address. Power on default is 0x0

Format:

- A  
Shows the current slave address
- AH nn  
Sets the high byte of the slave address, where nn is the high address in hex format.
- AL nn  
Sets the low byte of the slave address, where nn is the low address in hex format.

Example:

```
>a
>Slave Address: 0x0003
>ah 12
>al 4f
>a
>Slave Address: 0x124F
```

## 6.8 Command D

Command D sets an interval that is inserted between write cycles when command W or N is issued.

Format:

- D  
Shows current interval.
- D nn  
Sets the interval to nn, where nn is in hex format.

Example:

```
>d
>SPI write sequence delay: 0x0A
>d 34
>d
>SPI write sequence delay: 0x34
```

## 6.9 Command E

Command E enables the SPI write latch.

Format: > E

## 6.10 Command Q

Command Q writes to the status register if any.

Format:  
Q nn, where nn is the new status register value, which is in hex format.

## 6.11 Command M

Command M transfers the SPI application menu to Main menu.

Format: > M

## 6.12 Command L

Command L toggles the address mode between 16 bit and 8 bits. This setting determines whether 16-bit address or 8-bit address is used when write and get data from SPI device. Please read your SPI device datasheet to decide which mode should be used. 16-bit address mode is used by default.

Format: >L

## 6.13 Command '?'

Command '?' will display the entire SPI application menu.

Format: > ?

## 6.14 Command X

Command x writes all user input data to the SPI slave. There is NO Write enable latch command inserted between data.

Address bytes are not affected by this command.

Format: > x 12, 4d

Hex data 0x12 and 0x4d will be sent to SPI bus.

# 7 Revision History

Revision	Date	Description
1.0	3-2004	Initial release
1.1	5-2005	Added description for new pins layout to external target Removed command A in the I2C menu and add the list for the error codes. Added command x in the SPI menu for raw data write.