



NBPKBU-485CR Quad UART RS-485 Blade

Programming Reference Guide

Revision 1.0
April 2, 2009
Released

Table of Contents

1.	Introduction	3
2.	Installation.....	3
3.	Hardware	4
4.	Connector Pinouts	4
5.	Application Programming Interface	5
5.1	Serial Port Numbering Convention	5
5.2	Baud Rate Calculation	5
5.3	Open a Serial Port	6
5.4	Close a Serial Port.....	7
5.5	Software Flow Control	8
5.6	Half-Duplex or Full-Duplex Mode.....	9
5.7	Send Break.....	10
5.8	Red and Green LED Toggling	11

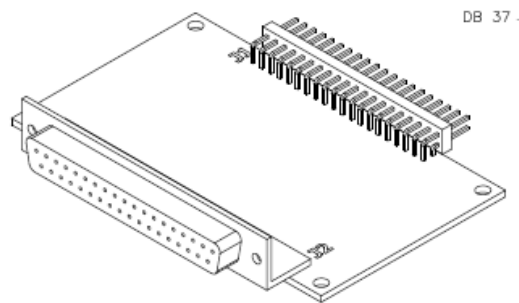
1. Introduction

The NBPKBU-485CR is a personality blade board for the NBP70EX-100CR. A quad UART and RS-485 level shifters provide four serial input/output channels.

2. Installation

The NBPKBU-485CR is mounted inside the PK70 enclosure. It has two connectors: a DB37 (J2) that connects to external devices, and a dual-row, 40-pin right-angle header (J1) that connects the NBPKBU-485CR to the PK70 interface connector.

To install the NBPKBU-485CR, remove the PK70 cover, plug the 40-pin J1 header into the 40-pin socket on the PK70, and install the four 4-40 mounting screws. Finally, replace the PK70 cover and cover screws.

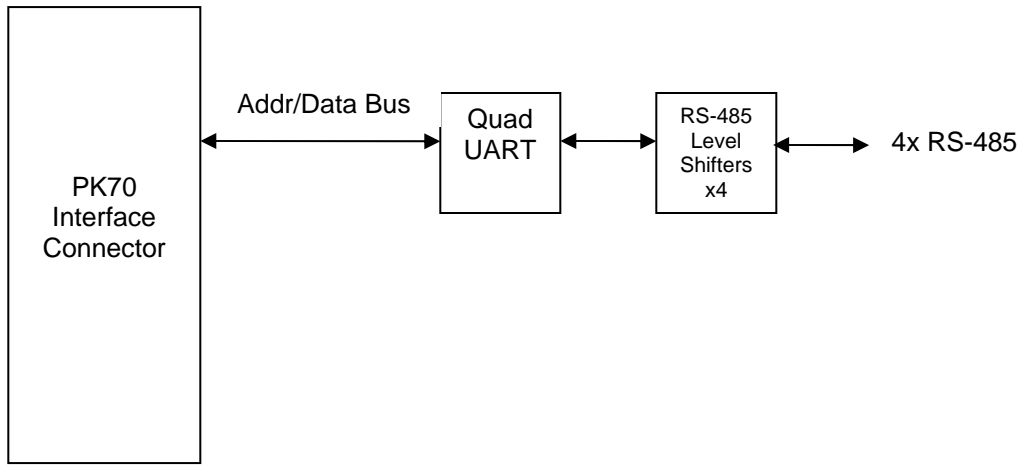


The software libraries are automatically installed with your PK70 development kit.

3. Hardware

The NBPKBU-485CR board interfaces to the PK70 through the 40-pin interface connector. This interface connector includes the address bus, data bus, QSPI, I²C, clock, reset, chip selects, and interrupt signals. The blade board uses the parallel address/data bus to interface to an Exar XR16L784CV-F quad UART.

Block Diagram



4. Connector Pinouts

The NBPKBU-485CR has a DB37 female connector, with a pinout designed to follow the ribbon cable style DB9 male crimp type connectors. Please refer to section 10.7 of the NBPKBU-485CR User's Manual, located in:

`\Nburn\docs\platform\PK70\ NBPKBU-485-UsersManual.pdf`

for the quad UART RS-485 cable signal configuration between the DB37 and four DB9 ports. For the pinout of the 40-pin J1 header that connects to the PK70 module, refer to the PK70 Hardware Manual (NBPK70.pdf), located in the same directory as the NBPKBU-485CR User's Manual.

5. Application Programming Interface

The following functions enable programming of the NBPKBU-485CR. The source code for this library are located in the \Nburn\PK70\include\NBPKQuad485.h and \Nburn\PK70\system\NBPKQuad485.cpp.

5.1 Serial Port Numbering Convention

The serial port numbering can be a bit confusing when looking at the hardware or schematics versus the software drivers. The ports are numbered from 1 to 4 when referencing hardware, including the quad UART cable. To maintain software conventions with other NetBurner serial drivers, the ports are numbered from 0 to 3 in the software drivers.

5.2 Baud Rate Calculation

When you specify a baud rate in the PK70Quad485OpenSerial() function, the baud rate for each port is calculated based on the quad UART crystal frequency as shown below:

$$\text{divider} = 14745600 / (\text{baud} * 16)$$

For example, the divider for a baud rate of 115,200 bits per second is eight. You can achieve any baud rate with a whole number divider value.

5.3 Open a Serial Port

Description:

Opens a serial port and returns a file descriptor if successful.

Syntax:

```
int PK70Quad485OpenSerial( int portnum, unsigned int baudrate,  
                           int stop_bits, int data_bits, parity_mode parity );
```

Parameters:

Parameter	Type	Description
portnum	int	UART to open; valid values are 0-3.
baudrate	unsigned int	Baud rate in bits per second.
stop_bits	int	Number of stop bits; valid values are 1 and 2.
data_bits	int	Number of data bits; valid values are 5-8.
parity	parity_mode	Valid values are eParityNone, eParityOdd, and eParityEven.

Returns:

Value	Description
(fd > 0)	File descriptor associated with the opened serial port if successful.
-1	SERIAL_ERR_NOSUCH_PORT
-3	SERIAL_ERR_PORT_ALREADYOPEN
-4	SERIAL_ERR_PARAM_ERROR (returned if stop/data bit value or parity mode is invalid)

A simpler version of this function called PK70Quad485SimpleOpenSerial() is also available. The only parameters required are the port number and baud rate. The stop bits, data bits, and parity are automatically set to 1, 8, and eParityNone, respectively.

```
int PK70QuadSimpleOpenSerial( int portnum,  
                              unsigned int baudrate );
```

5.4 Close a Serial Port

Description:

This function closes a serial port that is currently open.

Syntax:

```
int PK70Quad485SerialClose( int portnum );
```

Parameters:

Parameter	Type	Description
portnum	int	UART to close; valid values are 0-3.

Returns:

Value	Description
0	UART successfully closed.
-1	SERIAL_ERR_NOSUCH_PORT
-2	SERIAL_ERR_PORT_NOTOPEN

5.5 Software Flow Control

Description:

These functions enable and disable the sending (Rx flow control) or acknowledgement (Tx flow control) of XON/XOFF flow control characters.

Syntax:

```
void PK70Quad485SerialEnableTxFlow( int port, int enab );  
void PK70Quad485SerialEnableRxFlow( int port, int enab );
```

Parameters:

Parameter	Type	Description
port	int	UART to enable/disable flow control on; valid values are 0-3.
enab	int	'0' disables flow control; any non-zero value will enable it.

Returns:

Nothing to return.

5.6 Half-Duplex or Full-Duplex Mode

Description:

Calling either the half-duplex or full-duplex function will enable half-duplex or full-duplex mode, respectively. One of these functions must be called after opening the serial port.

Syntax:

```
void PK70Quad485SerialSetFullDuplex485( int port );  
void PK70Quad485SerialSetHalfDuplex485( int port,  
    BOOL bEcho = FALSE );
```

Parameters:

Parameter	Type	Description
port	int	UART to enable half or full-duplex on; valid values are 0-3.
bEcho	BOOL	Setting to "FALSE" disables echo; setting to "TRUE" enables echo. This parameter only applies to the half-duplex enabling function. If this parameter is not explicitly provided in the function call, echo is disabled by default.

Returns:

Nothing to return.

5.7 Send Break

Description:

Sets a break in the UART transmission for a given period of time.

Syntax:

```
void PK70Quad485SerialSendBreak( int port, DWORD time );
```

Parameters:

Parameter	Type	Description
port	int	UART to set the break transmission on; valid values are 0-3.
time	DWORD	The length of time that the transmission break will occur, in ticks per second (20 ticks = 1 second by default).

Returns:

Nothing to return.

5.8 Red and Green LED Toggling

Description:

These functions toggle the pair of red and green LEDs for each UART.

Syntax:

```
void PK70Quad485SerialRedLed( int port, bool on );  
void PK70Quad485SerialGreenLed( int port, bool on );
```

Parameters:

Parameter	Type	Description
port	int	UART whose red/green LED will be toggled; valid values are 0-3.
on	bool	Setting to “true” turns on the LED; setting to “false” turns off the LED.

Returns:

Nothing to return.