SPI2UART v2 Datasheet

Product Brief

Provides a bridge between an embedded SPI bus and up to four buffered UART connections. Can also be used for RS232, RS485, DMX512, Modbus.

This board provides up to four buffered UART channels from a single SPI connection. If more than four UARTs are required, then a second board can be added to provide eight channels and so on.

Each UART peripheral has a 1000 byte buffer dedicated for receiving and another 1000 byte buffer dedicated to transmitting allowing a large amount of data to be sent and received simultaneously which should mean that you never lose a byte.

The LED glows dimly when the board is powered, glows at half power when transmitting or receiving via a UART and glows at full power when communicating via SPI.

The board requires a 3.3V supply to function but the SPI and UART RX pins can accept both 5V and 3.3V inputs. The UART TX pin will output at 3.3V which should be compatible with most if not all 3.3V and 5V devices directly without the need for any additional voltage level shifting circuitry.

A MAX2323 chip can be used to convert the logic level RX/TX signals to RS232 levels.

An Arduino library and Flowcode component are available to help drive this hardware. Also provided is the module firmware allowing you to ensure you are always up to date. Please note a PICkit 3 will be required to reprogram the on-board microcontroller

The version 2 board features a new on-board microcontroller in a reduced pin package which is far easier to manufacture, bringing up quality and reliability whilst helping to reduce wastage and bringing down the cost. The v2 board is 100% compatible with the v1 board.

SPI Commands

The command code and UART channel are packed together into a single byte to increase efficiency. The command code resides in the top 4 bits and the channel resides in the bottom 2 bits.

Here are the bits in the byte and their representation: 0b**cccc**00**uu**
Where **cccc** is the command code and **uu** is the UART channel (0 - 3).
Channel 0 controls pins TX1 and RX1, channel 3 controls pins TX4 and RX4

Read Receive Buffer Size

Description	Reads the number of bytes in the selected channel receive buffer.
Command Code	0x10 – 0x13
Parameters	N/A
Returns	NumBytes (0-255)

If the returned value NumBytes is 255 then this means that the number of bytes stored in the buffer is 255 or more. It may be worth calling this function again once 255 bytes have been read from the buffer to see if all values have been retrieved.

Read Receive Buffer

Description	Reads data byte(s) from the selected channel receive buffer.
Command Code	0x20 – 0x23
Parameters	NumBytes (0-255)
Returns	DataByte (0-255) [0- NumBytes]

Before calling this command you must first call the Read Receive Buffer Size function to be sure of the max number of bytes to read.

Read Transmit Buffer Size

Description	Reads the number of bytes in the selected channel transmit buffer.
Command Code	0x30 – 0x33
Parameters	N/A
Returns	NumBytes (0-255)

If the returned value NumBytes is 255 then this means that the number of bytes stored in the buffer is 255 or more. It may be worth calling this function again once enough time for 255 bytes to be transmitted at the selected baud rate from the buffer to see if the buffer is empty.

Write Transmit Buffer

Description	Puts data byte(s) into the selected channel transmit buffer.
Command Code	0x40 – 0x43
Parameters	NumBytes (0-255) DataByte (0-255) [0- NumBytes]
Returns	N/A

Set Channel Baud Rate

Description	Sets the selected channel baud rate
Command Code	0x80 – 0x83
Parameters	Baud (0-7)
Returns	N/A

Here are the options for the baud rate parameter.
0=1200, 1=2400, 2=4800, 3=9600, 4=19200, 5=38400, 6=57600, 7=115200

Channel baud rate changes are stored into the device's none volatile memory and the last assigned value will be automatically loaded after a power cycle. Default baud rate from factory settings is 9600 for all channels.

Examples

To read the number of bytes in UART channel 0 receive buffer we send the following command code and then perform a read.

```
CS Low
SPISend ( 0x10 )
NumBytes = SPIReceive ( 0xFF )
CS High
```

To read 5 bytes from UART channel 1 receive buffer we send the following command code, the number of bytes and then perform enough reads to pull out all the data we want.

```
CS Low
SPISend ( 0x21 )
SPISend ( 0x05 )
Byte[0] = SPIReceive ( 0xFF )
Byte[1] = SPIReceive ( 0xFF )
Byte[2] = SPIReceive ( 0xFF )
Byte[3] = SPIReceive ( 0xFF )
Byte[4] = SPIReceive ( 0xFF )
CS High
```

To write 5 bytes to UART channel 2 transmit buffer we send the following command code, the number of bytes and then perform enough writes to send out all the data we want.

```
CS Low
SPISend ( 0x42 )
SPISend ( 0x05 )
SPISend ( Byte[0] )
SPISend ( Byte[1] )
SPISend ( Byte[2] )
SPISend ( Byte[3] )
SPISend ( Byte[4] )
CS High
```