SPI Bus with BL233 and I2C2PC

1. Introduction

The BL233B was primarily designed to interface to I2C bus devices. I2C is a thoroughly standardised protocol.

SPI however is a more generic term covering a multitude of interfaces from a variety of manufacturers, that only really have the following in common:

- Synchronous serial protocol with a clock line, and no special start or stop conditions
- Chip_Select or STroBe pin controls framing

These features vary between chips and manufacturers

- CS/STB pin can have either polarity and change state in varying places during the cycle
- Some devices do simultaneous data input and output, and so require separate Din and Dout pins
- Some have one bi-directional data In-Out pin and others have separate ones.
- Many devices with separate Din/Dout pins can have them connected together
- most transfer data in multiples of 8 bits, but some don't
- some clock data on the rising edge, and some on the falling edge. Others do both depending on the state of the clock line when CS changes

There are two trademarks commonly used for buses of these types: Microwire and SPI

Contents

2. I2C2PC: Set Jumpers

Bus#2,3 of I2C2PC adaptor are configured with pin 5 as IRQ-IN, from the factory. *You must change jumpers before using CS*.

Bus#1 does not have CS capability.

The jumpers are labelled on the silkscreen of the pcb. J5 for Bus 2, J4 for Bus 3.

You might want to write CS on the box by the bus number for future users.

see:I2C2PC Datasheet http://www.i2cchip.com/pdfs/i2c2pc.pdf

3. Commands

Commands are fully explained in the BL233B Datasheet http://www.i2cchip.com/pdfs/bl233_b.pdf. See Section 10:SPI . The datasheet is the detailed reference tom commands.

The **Y** command is used for SPI data transfers. It defaults to 8 bit transfers, but can do shorter transfers i.e. 1-8 clock pulses.

4. Clock Pin

4.1. Clock Phase and Clock Polarity

The common SPI interface hardware in a microprocessor results in the data pins changing or being sampled at the instant of the clock edge. This means that the slave must use the other clock edge, for valid data transfer.

BL233 does not change data and clock at the same time. This means that the clock phase is not important. Either clock phase will work.

4.2. Timing

The J command changes bit timing for SPI, the same as for I2C.

5. CS Pin

Note that when supplied from the factory, Bus#2,3 of I2C2PC adaptor are configured with pin 5 as IRQ-IN. You must change jumpers before using CS-out. Bus#1 does not have CS capability.

The CS pin is also known as STroBe or Slave Select.

You control the CS pin directly using the "O" command. The "O" command controls all the pins of the BL233 so you have to use a bit mask that changes the pin you want, and not the others. (ie set this wrong, and you will disable the data and clock pins)

Commonly you would set the CS before the transfer, and reset it at the end of the transfer.

	CS2	*CS3*	*Command*
Initialise	1	1	**`P030CF`**
set	0	0	**`O00`**
set	1	0	**`O10`**
set	0	1	**`O20`**
set	1	1	**`O30`**

6. Data Pins

Data pins are known as Data In, Data Out, SDI, SDO, MOSI, MISO.

Normally the BL233 has a single SDA pin for Din and Dout. Many SPI devices have two separate pins for this MOSI[1] and MISO[2]. SPI devices are often designed to have common MISO & MOSI pins for multiple devices, and so the MISO pin is tristateable. In many cases they can be connected together (to the common SDA) as the slave does not read and write data at the same time. You can establish this from the datasheet, especially by looking at the datasheet, to see the tri-state timing of the MISO pin, to see if it overlaps any of the incoming data on the MISO pin.

[1] Master Out Slave In, also referred to as Dout here
[2] Master-In Slave-Out , also referred to as Din here

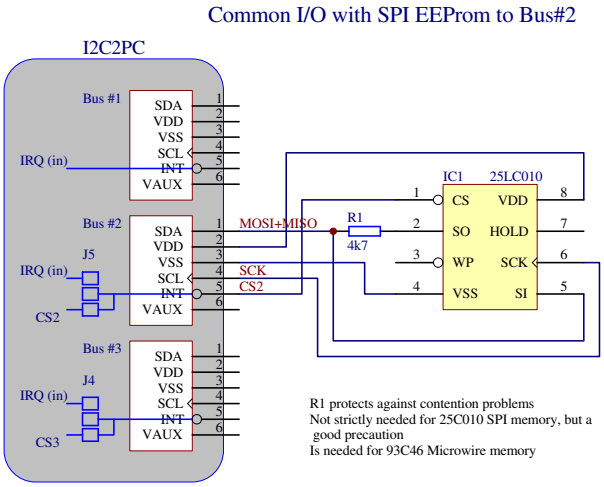

Figure 1:

As an example of this the common SPI EEProm 25LC010 and Microwire EEProm 93C46 will work (see contention below for 93C46),

In this case you will use either Bus2 or Bus3 and the CS pin will be pin 5 (CS), and the SPI bus will be able to use a single bus connector

6.1. Avoiding Contention Issues

Some devices can have some bus contention, that does not affect correct operation. It may however cause excess current pulses. To avoid this it is recommended that the MOSI pin is connected direct to SDA, and the MOSI pin (data out of the slave) is connected via a 4k7 resistor to SDA. This limits the contention current to 1mA, and prevents any problems.

6.2. Separate Din/DOut: Half Split Buses

In some cases however you require separate lines for Din/Dout. In this case you must select one of the half-split or full split buses using the G command. The best choices are the half-split buses G9: Bus2 Half Split, or GA: Bus3 Half Split.

	SCLK	MISO/SDAIn	MOSI/SDAOut	CS	CS=0	CS=1
G9 Bus2 Half Split CMOS	Bus2, pin4 (SCL)	Bus2, pin5 (CS)	Bus2, pin1 (SDA)	Bus3,pin5 (CS3)	`O10`	`O30`
GA Bus3 Half Split TTL	Bus3,pin4	Bus3,pin5 (CS)	Bus3,pin1 (SDA)	Bus2,pin5 (CS2)	`O20`	`O30`

When you use a half split bus, the data and clock signals are on one bus, and the CS pin/s will be on another.

For example you can use Bus 2 for clock and data, and the three pins of Bus 3 as three CS pins.

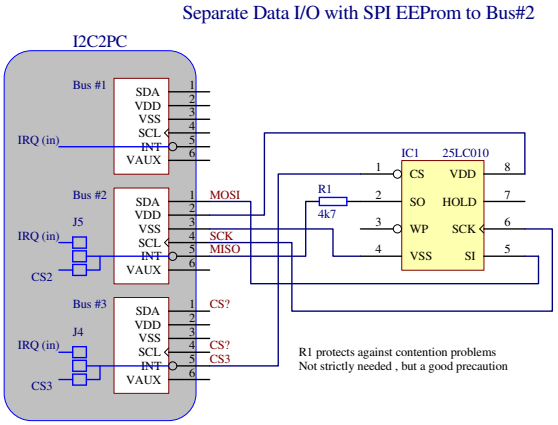

Figure 2:

Using the AUX pin for SDAIn/MISO

The AUX pin (pin6) is unused on the adaptor and free for user applications. You might find it convenient to modify the adaptor, and thus have the whole SPI bus on a single connector.

You have two options: Keep CS on pin 5 (add two wires), or just use 1 wire onto the AUX pin, eg using Bus 2, solder a wire link from J4-CS to Bus2-AUX (pin6).

#	Keep Jumper	Keep CS on pin5
1	SDAOut, MOSI	SDAOut, MOSI
2	V+	V+
3	GND	GND
4	SCL	SCL
5	SDAin, MISO	CS (unplug jumper, add wire)
6	CS (addwire)	SDAin, MISO (add wire)

7. Examples

The examples below have not been tested.

7.1. Microwire EEProm 93C46A

Note that 93C46A is 8 bit part, not to be confused with 93C46B which is 16 bit.

The 93C46 works with common DI and DO pins, but needs a resistor from DO to SDA. see 6.1 on the preceding page

CS will be CS3. Jumper J4 must be set to CS on the I2C2PC.

The first byte written contains a start bit (always 1), Opcode (2 bits), address (6 bits).So to read 2 bytes from A0:

```
G3 //select bus 3
PO30CF O10 //initalise CS pins, set CS3 low
O30 //set CS3 High
Y //enter SPI mode
W C0 //Opcode read, address=0
R 02 //read 2 bytes
O10 //set CS3 low
```

Lets try again, but using separate DI and DO pins as an exercise. CS will be CS2 this time as CS3 is being used for data in. J4 and J5 must be set to CS

```
GA //select half split bus 3. CS2 will be the CS pin used
PO30CF O20 //initalise CS pins, set CS2 low
O30 //set CS2 High
Y //enter SPI mode
W C0 //Opcode read, address=0
R 02 //read 2 bytes
O20 //set CS2 low
```

A. Revision History

Rev	Date	Changes
0	24 Sep 07	First Release
1	29 Jul 15	Update format, minor changes
2		
3		

Nomenclature

MISO Master-In Slave-Out

MOSI Master Out Slave In: data pin of SPI serial device