CML Microcircuits

COMMUNICATION SEMICONDUCTORS

Application Note

C-BUS to SPI interfacing

This document provides techniques for interfacing the CML proprietary C-BUS to an SPI bus. It explores the easier handling enabled by modern implementations of SPI ports and delves into some of the issues and trade-offs that arise. The introduction is not meant to serve as a tutorial to SPI and the material covered is only mentioned where it is relevant to the operation of C-BUS. Readers are expected to have a good knowledge of SPI and are advised to carefully study the SPI implementation that they plan to use.

1 Introduction to SPI hardware

The SPI bus specifies four logic signals:

1. SCLK - Serial Clock (output from master)
2. MOSI - Master Output, Slave Input
3. MISO - Master Input, Slave Output
4. SS - Slave Select (active low output from master, input to slave)

Alternative naming conventions are also widely used:

1. SCK - Serial Clock
2. SDI, DI, SI - Serial Data In
3. SDO, DO, SO - Serial Data Out
4. nCS, CS, nSS, STE - Chip Select, Slave Transmit Enable

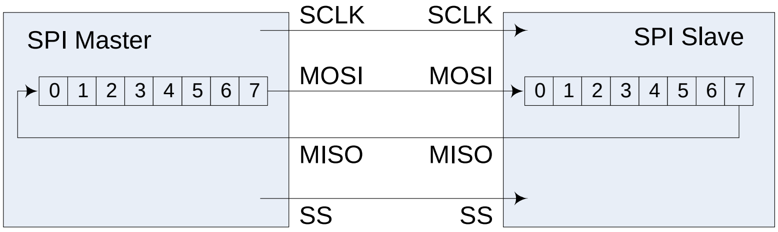


Figure 1. Typical basic SPI connection

A typical connection between two SPI devices is shown in Figure 1. The SPI bus operates with a single master and one or more slaves. The SS is effectively a chip select on a slave and, if a single slave is used, it may be possible to wire its SS pin to logic 0. Some slaves, however, require a falling edge on their SS input to initiate any actions. With multiple slaves, an independent SS signal is required from the master for each slave. Some processor's SPI ports provide more than one SS output when bus mastering but often GPIO pins, under software control, must be committed to this function. Multiple slave connection is illustrated in Figure 2.

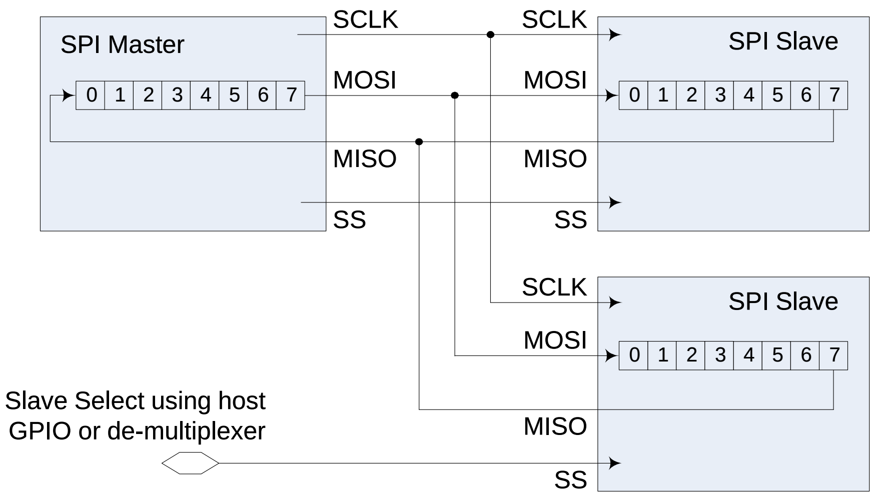


Figure 2. Connecting multiple slaves to one master

Most SPI devices have tri-state outputs that go high impedance when a slave is not selected. This is a requirement for bus sharing unless some form of external multiplexing is used. Some SPI devices permit multiple slaves to be selected and provide an arbitration or addressing system in the data transfer.

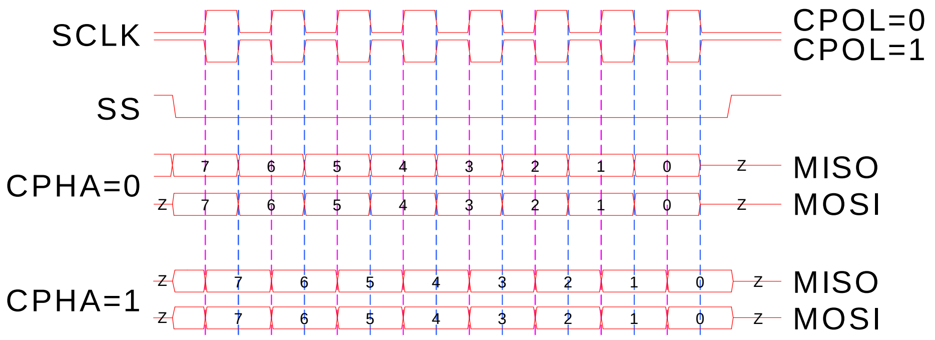


Figure 3. Relative timing for all SPI modes

SPI permits control of the polarity and phase with respect to data transfers. The settings of these parameters determine the ‘Mode’ of the SPI port and there are typically 4 such mode settings. The timing of the relative signals is shown in Figure 3 for all modes. The modes are tabulated in Table 1.

Clock Polarity - CPOL	Clock Phase - CPHA	SPI Mode
0	0	0
0	1	1
1	0	2
1	1	3

Table 1. SPI Modes

Many modern implementations of SPI have discarded the looping register approach and use a configuration similar to that illustrated in Figure 4. Data transmission become independent from reception and this has several advantages:

- Transmitting data does not cause garbage to be looped back into the receiver
- The formatting of the data is under host control
- The SPI transaction becomes readily configurable

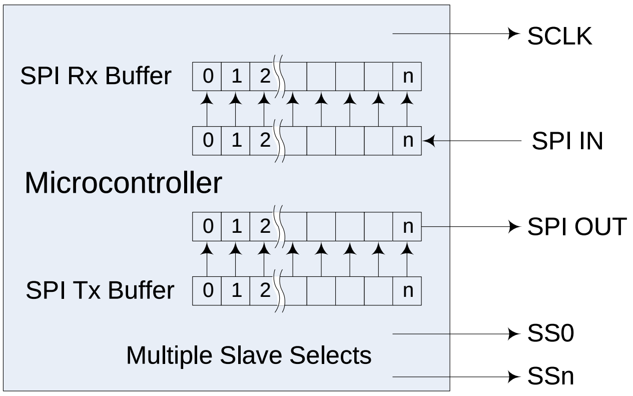


Figure 4. SPI implementation on a modern microcontroller

Modern implementations also permit extensive control of the signal timing. For example:

- Autonomous control of SS
- Programmable duration from the falling edge of SS to the first clock edge
- Programmable duration between concatenated reads and writes
- Streaming modes that allow continuous transfer for a preset number of bytes
- FIFOs to reduce software load

2 Connecting C-BUS to SPI

C-BUS is very similar to SPI and in most cases, can be connected directly to an SPI master. CML devices are always connected as slaves. Multiple CML devices can be connected by multiplexing SS since the output equivalent to MISO, Reply Data on C-BUS, is tri-state when SS is high. The connection method is illustrated in Figure 5.

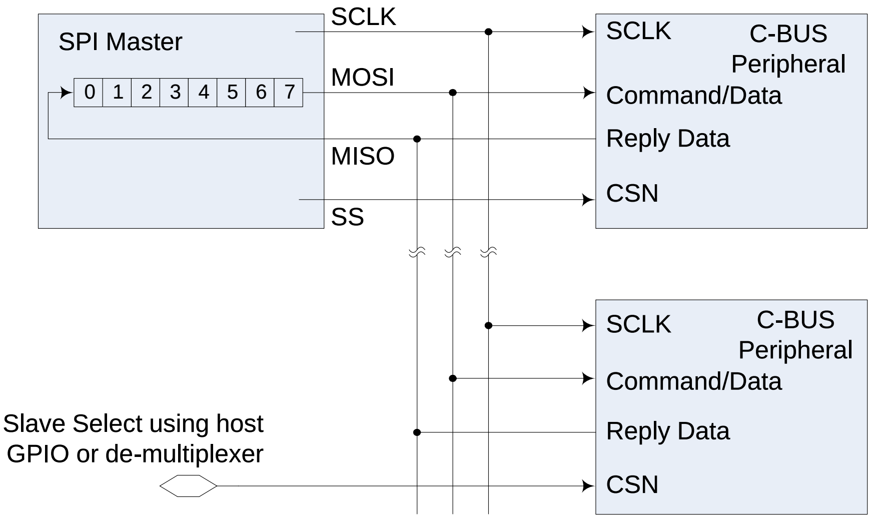


Figure 5. Typical connection for one or more C-BUS slaves

C-BUS transactions are initiated by a falling edge on CSN but this must not rise again until the full C-BUS transaction is completed. C-BUS transactions are normally 8, 16 or 24 -bit. Some of the later devices also have a streaming mode in which CSN must be asserted for the full duration of the data transfer. Further information on C-BUS transactions can be found in the Application Notes section of the CML website.

The following sections discuss ways of configuring and handling SPI when interfacing C-BUS. Please see Table 2 at the end of this document for the various naming conventions used in C-BUS.

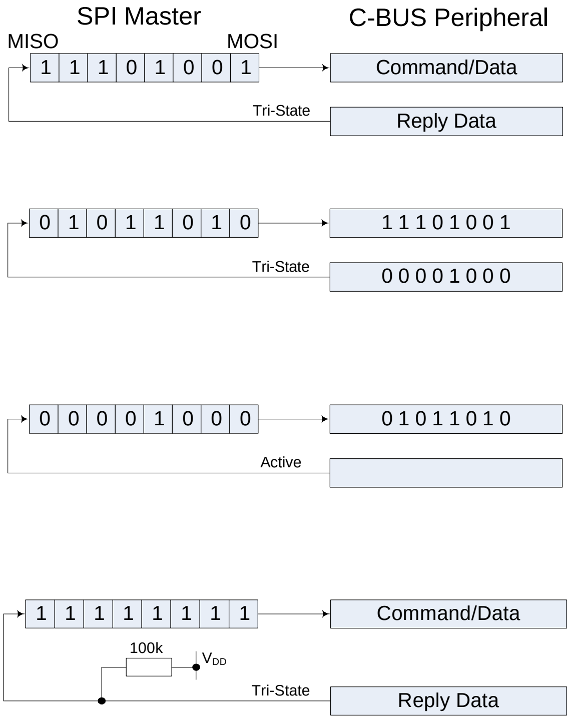
2.1 Basic SPI port connection

The basic SPI port comprises a single 8-bit register with access at either end such that the data can be looped through the device as illustrated in Figure 1. For this type of implementation:

1. Configure the SPI port as a master.
2. Program the appropriate clock rate
3. The clock phase and polarity should be set so that data is transmitted on the falling edge and received on the rising edge. Normally SPI mode 3 will work but check the relative timing of SS low to the change of state on SPI Clock. There must be a delay that meets the requirements of C-BUS. Alternatively, use a general-purpose output pin for C-BUS Chip Select and control it in software.
4. The MOSI pin should be connected to the C-BUS Command/Data input.
5. MISO should be connected to the C-BUS Reply Data output.
6. The SPI Clock should be connected to the C-BUS Serial Clock input.
7. SS (Slave Select) should be connected to C-BUS Chip Select. Essentially, this can be considered as a device chip select that must be active low for the full duration of any C-BUS transaction.

2.2 Issues with SPI register looping

The Command/Data and Reply Data lines of C-BUS are not looped. If a read of C-BUS is required, then a command is sent into Command/Data and the response is subsequently read back from Reply Data. Reply Data is tri-state until after the Command/Data byte has been received by the C-BUS peripheral. Following this, after the required inter-byte duration, Reply Data becomes active and Command/Data is expected to remain at the current level. On the SPI master both MOSI and MISO are active due to the looping effect of the register. This can cause problems when C-BUS transactions are used that returns bytes (such as a read of the Status Register). Here the Command/Data byte (clocked out on MOSI) will cause the master's SPI register to fill with 'random' data from MISO. The following read from C-BUS Reply Data on MISO, will cause the 'random' data in the SPI register to loop onto C-BUS Command Data. When the C-BUS CSN is taken high to terminate the transaction, the 'random' data clocked into Command/Data may cause unexpected operation in the C-BUS peripheral. To avoid any problems, the C-BUS 'Reply Data' should be pulled high or the SPI contents cleared to 0xFF between data transfers. This scenario is illustrated in Figure 6.

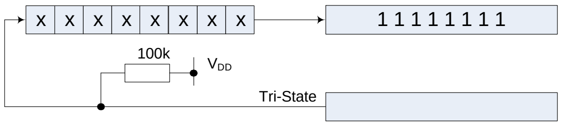


6a. A C-BUS command has been loaded into the SPI register and a transmit task is activated. As the C-BUS command is transmitted, the level at MISO is read in as an arbitrary value.

6b. The C-BUS command has been clocked into the C-BUS peripheral and the reply is loaded ready to send. The SPI register now contains a 'random' byte.

6c. The Reply Data has been clocked back into the SPI register via MISO and the 'random' byte has been clocked into C-BUS Command Data. This may cause unexpected operation in the C-BUS peripheral.

6d. Using a 100k pull-up on MISO causes 0xFF to be looped into the SPI register instead of random data.
Alternatively, the SPI register could be loaded with 0xFF prior to all SPI receive tasks.



6e. 0xFF is clocked into Command/Data instead of 'random' data. This is normally safe for all C-BUS devices.

Figure 6. Avoiding problems with data looping through the SPI register

2.3 Autonomous control of SS

It is conventional on most SPI implementations to find that SS can be handled without software intervention - autonomous control of SS. This feature may be a default mode or it may have to be selected. Autonomous control of SS will require that a default pin is used for the SS signal or a limited number of options will be available. If the C-BUS timing prevents SS autonomous control, then it must be disabled and a GPIO pin used instead. In this case:

1. The GPIO pin must be set to low
2. SPI transfer initiated
3. On completion signal from SPI the GPIO pin must be set high

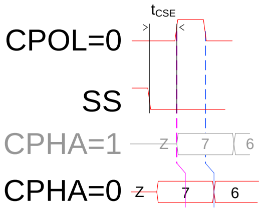
It is also necessary to ensure that the GPIO pin is not held low during start-up or initialisation following application of power.

2.4 Using multiple byte SPI transactions

Many versions of SPI permit a preset number of bytes to be transferred in any one transaction (the period during which SS is asserted). Caution must be used with this mode to ensure that the timing requirements of C-BUS are met.

2.5 Considerations for C-BUS timing

C-BUS timing parameters should be obtained from the C-BUS Timings section of the RELEVANT device's data sheet.



It is possible to use SPI Mode 0 if the duration from SS low to SCLK first edge (high in this case) is programmable. This duration must be programmed to be equal to or longer than t_{CSE} , given in the C-BUS timing parameters. This is only of benefit where it is required that SCLK be low between SPI transactions, for example when another slave on the bus requires this.

Figure 7. Using SPI Mode 0

The SPI SCLK normally has an equal duty cycle which is ideal for C-BUS. However, if approaching the maximum C-BUS clock cycle time, then confirm that the duration of the shortest clock edge is equal or longer than t_{CSE}, t_{CH}, t_{CL}, t_{CSH}. Also, confirm that the clock period is equal to or longer than t_{NXT} if multi-byte SPI transactions are used.

C-BUS transactions cannot be concatenated. The delay between successive C-BUS transactions is given by t_{CSOFF} and it is usually of longer duration than the clock cycle time t_{CK}. It is likely that a timer or some other mechanism will be required to prevent this timing parameter being violated.

If the highest burst rate (highest clock speed) is not required from C-BUS, then make the SPI SCLK period;

1. longer than tNXT if the time between transactions is controlled or
2. longer than tCSOFF for the simplest timing control.

3 C-BUS signal pin names

C-BUS has always comprised 5 signal lines but these have been given many alternative names. The following table gives these alternatives, but groups them as they are found on particular C-BUS devices. The entries are listed historically, with the latest naming convention at the beginning of the table. The table is headed by the names used in this document.

SClk	Command/Data	Reply Data	CSN	IRQN
Serial Clk	CMD Data	Reply Data	CSN	IRQ
SCLK	CMD Data	Reply Data	CSN	IRQ
CLK	CData	RData	CSN	IRQN
SCLK	CData	RData	CSN	IRQN
SCK	SDI	SDO	CSN	-
Serial_Clock	CMD_Data	Reply_Data	CSN	IRQN
Serial Clock	CMD Data	RPLY Data	$\overline{\text{CS}}$	$\overline{\text{IRQ}}$
SCLK	CMD	RPLY	CSN	IRQN
Serial Clock	Command Data	Reply Data	$\overline{\text{CS}}$	$\overline{\text{IRQ}}$
SERCK	COMDATA	REPDATA	CSN	IRQN

Table 2. Alternative names for C-BUS signal pins.

This list is not exhaustive and other variations of these names will occasionally be found in older CML product.

CML does not assume any responsibility for the use of any algorithms, methods or circuitry described. No IPR or circuit patent licenses are implied. CML reserves the right at any time without notice to change the said algorithms, methods and circuitry and this product specification. CML has a policy of testing every product shipped using calibrated test equipment to ensure compliance with this product specification. Specific testing of all circuit parameters is not necessarily performed.

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS Singapore	China
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 67450426 **Fax:** +65 67452917 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com	**Tel:** +86 21 6317 4107 +86 21 6317 8916 **Fax:** +86 21 6317 0243 **Sales:** cn.sales@cmlmicro.com.cn **Tech Support:** sg.techsupport@cmlmicro.com
- www.cmlmicro.com -			