

UCD30xx

SPI Module

Programmer's Manual

Literature Number: xxxxxx

Date

Table of Contents

UCD30xx	1
SPI Module	1
Programmer's Manual	1
Literature Number: xxxxxx.....	1
Date	1
1 SPI Module Overview.....	3
2 SPI Operation.....	3
2.1 SPI Operation; Three-Pin Option.....	3
2.2 SPI Operation; Four-Pin Option	3
2.3 Data Format	3
2.4 Clocking Modes	3
2.5 Baud Rate Limitations	3
3 SPI – Serial Peripheral Interface.....	3
3.1 SPI Control Register 1 (SPICTRL1).....	3
3.2 SPI Control Register 2 (SPICTRL2).....	3
3.3 SPI Control Register 3 (SPICTRL3).....	3
3.4 SPI Shift Register 0 (SPIDAT0)	3
3.5 SPI Shift Register 1 (SPIDAT1)	3
3.6 SPI Receive Buffer Register (SPIBUF)	3
3.7 SPI Pin Control Register 1 (SIPPC1).....	3
3.8 SPI Pin Status Register 1 (SPIST1)	3
3.9 SPI Pin Control Register 3 (SIPPC3).....	3
3.10 SPI Pin Control Register6 (SIPPC6).....	3
4 Useful C statements	3

1 SPI Module Overview

The SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (3 to 32 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the UCD3xxx and external peripherals or another microcontroller. Typical applications include interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, and analog-to-digital converters.

Communication is carried via 3 pins. SPI_CLK is driven by SPI master and is the clock signal that synchronizes the data shifting in or out. SPISIMO is the data output signal and carries the outgoing information from UCD3xxx to the connected device. SPISOMI pin is receiving the ingoing data information from the transmitting device.

The SPI has the following attributes:

- 32-bit shift register
- Receive buffer register
- 8-bit baud clock generator
- Serial clock (SPI_CLK) I/O pin
- Slave in, master out (SPISIMO) I/O pin
- Slave out, master in (SPISOMI) I/O pin

Note: In some documents including UCD3xxx pin out drawings and schematics the pins SPISIMO and SPISOMI are alternatively called SPI_DO and SPI_DI respectively.

The SPI allows software to program the following options:

- SPISOMI/SPISIMO pin direction configuration
- SPI_CLK pin source (external/internal)
- SPI_CLK frequency (interface clock [ICLK] /2 through /256)
- SPI pins as functional or digital I/O pins
- Character length (3 to 32 bits)
- Phase (delay/no delay)
- Polarity (high or low).

2 SPI Operation

2.1 SPI Operation; Three-Pin Option

Figure 1. SPI 3-pin Operation



Data written to the shift register (SPIDAT0) initiates data transmission on the SPISIMO pin, most significant bit (MSB) first. Simultaneously, received data is shifted through the SPISOMI pin into the least significant bit (LSB) of the SPIDAT0 register. When the selected number of bits has been transmitted, the data is transferred to the SPIBUF register for the CPU to read. Data is stored right-justified in SPIBUF.

When the specified number of bits has been shifted through the SPIDAT0 register, the following events occur:

- The RXINTFLAG bit (SPICTRL3.0) is set to 1
- The SPIDAT0 register contents transfer to the SPIBUF register
- An interrupt is asserted if the RXINTEN bit (SPICTRL3.1) is set to 1

In slave mode configuration (MASTER = 0 and CLKMOD = 0), data shifts out on the SPISOMI pin and in on the SPISIMO pin. The SPI_CLK pin is used as the input for the serial shift clock, which is supplied from the external network master. The transfer rate is defined by this clock.

Data written to the SPIDAT0 register is transmitted to the network when the SPI_CLK signal is received from the network master. To receive data, the SPI waits for the network master to send the SPICLK signal and then shifts data on the SPISIMO pin into the SPIDAT0 register. If data is to be transmitted by the slave simultaneously, it must be written to the SPIDAT0 register before the beginning of the SPI_CLK signal.

2.2 SPI Operation; Four-Pin Option

The three-pin option and the four-pin options of the SPI are identical in the master mode (CLKMOD = 1), except that the four-pin option uses either SPIENA or SPISCS pin. The I/O direction of these pins is determined by the CLKMOD control bit as SPI not general purpose I/O.

4-pin option with SPISCS

To use the SPISCS as an automatic chip select pin, the SPISCS pin must be configured to be functional (SPIPC6.4 = 1). In this mode, the master will drive this signal low when data has been written to SPIDAT1 and then drive the pin high

The following conditions apply for words with fewer than 32 bits:

- ❑ Data must be left-justified when it is written to the SPI for transmission
- ❑ Data is right-justified when read back from the receive register

The buffer contains the most recently received word, right-justified, plus any bits that are left over from previous transmissions that have been shifted to the left. The diagram below shows how a 14-bit word is stored in the buffer once it is received.

SPIBUF Bits

D31	D30	D29	D28		D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	0	1	-----	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

In transmit mode, the SPIBUF register contains the most recently transmitted word, left-justified. The diagram below shows how a 30-bit word needs to be written to the buffer in order to be transmitted correctly.

To allow for the efficient transmission of byte-sized words, if a character length is programmed for 8 bits or less, the SDPDAT [7] bit instead of SDPDAT[15] is the source of the serial out data. This prevents the need to further add 8 justification bits.

SDPDAT Bits

D31	D30	D29	D28		D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	-----	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	X	X

2.4 Clocking Modes

There are four clock modes in which SPI_CLK may operate, depending on the choice of the phase (delay/no delay) and the polarity (rising edge / falling edge) of the clock. When operating with PHASE active, the SPI makes the first bit of data available after the SPIDAT0 register is written and before the first edge of

SPI_CLK. The data input and output edges depend on the values of both POLARITY and PHASE as shown in [Table 2](#).

Table 2. Clocking Modes

POLARITY	PHASE	ACTION
0	0	Data is output on the rising edge of SPICLK. Input data is latched on the falling edge.
0	1	Data is output one half-cycle before the first rising edge of SPICLK and on subsequent falling edges. Input data is latched on the rising edge of SPICLK.
1	0	Data is output on the falling edge of SPICLK. Input data is latched on the rising edge.
1	1	Data is output one half-cycle before the first falling edge of SPICLK and on subsequent rising edges. Input data is latched on the falling edge of SPICLK.

2.5 Baud Rate Limitations

It is recommended to operate the master and slave SPIs at the same baud rate. However, when this is not possible the SPICLK ranges specified in [Table 3](#) must be followed to ensure proper data transfer. The SPICLK rate is set by adjusting the PRESCALE value in the SPICTRL1 register.

Table 3. SPICLK Ranges

POLARITY	PHASE	SPICLK RATIO
X	0	$\frac{MasterSPICLK}{2} \leq SlaveSPICLK \leq (MasterSPICLK + 1\%)$
X	1	$\frac{MasterSPICLK}{2} \leq SlaveSPICLK \leq (MasterSPICLK \times 2)$

In all clocking mode configurations, the slave SPICLK must never be less than half the speed of the master SPICLK. Doing so may allow the master to start a new SPI transmission before the slave is ready. When operating with PHASE = 0, the slave SPICLK must not be more than 1% faster than the master SPICLK. When operating with PHASE = 1 the slave SPICLK must not be more than two times faster than the master SPICLK. If the slave SPICLK exceeds the master SPICLK by more than 1% when PHASE = 0 or by 2x when PHASE = 1 there is a possibility that the slave will move data from the input shift register to SPIBUF before the master is finished transferring data.

3 SPI – Serial Peripheral Interface

SPI Registers have the following attributes:

- 8-bit wide
- Addresses placed on word boundaries
- Byte, half-word and word writes permitted
- All Registers have read/write access in any mode

3.1 SPI Control Register 1 (SPICTRL1)

Address FFF7F800

Bit Number	13:6	5:0
Bit Name	PRESCALE	CHAR_LEN
Access	R/W	R/W
Default	0000_0000	00_0000

Bits 13-6: PRESCALE – Baud Rate Configuration

00000000 = Baud Rate equals ICLK

00000001 = Baud Rate equals ICLK/2

00000010 = Baud Rate equals ICLK/3

.....

11111111 = Baud Rate equals ICLK/256

Bits 5-0: CHAR_LEN – These bits control how many times the SPI shifts per character transmitted or the number of bits per character. The binary value of the bit length must be programmed into this Register.

3.2 SPI Control Register 2 (SPICTRL2)

Address FFF7F804

Bit Number	5	4	3	2	1	0
Bit Name	CLK_MOD	SPI_ENA	MASTER	RESERVED	POLARITY	PHASE
Access	R/W	R/W	R/W	-	R/W	R/W
Default	0	0	0	-	0	0

Bit 5: CLK_MOD – SPI Clock Mode

0 = SPI utilizes external clock (Default)

1 = SPI drives external clock

Bit 4: SPI_ENA – SPI Interface Enable

0 = Disables SPI Interface (Default)

1 = Enables SPI Interface

Bit 3: MASTER – Configures Master/Slave mode for SPI Interface

0 = Slave Mode (Default)

1 = Master Mode

Bit 1: POLARITY – Polarity of SPI Clock

0 = SPI Clock non-inverted when driven off chip (Default)

1 = SPI Clock inverted when driven off chip

Bit 0: PHASE – Configures phase of output clock

0 = SPI Clock is not phased from peripheral clock (Default)

1 = SPI Clock is phased from peripheral clock

3.3 SPI Control Register 3 (SPICTRL3)

Address FFF7F808

Bit Number	3	2	1	0
Bit Name	OVERRUN_INT_ENA	OVERRUN_INT_FLAG	SPI_INT_ENA	SPI_INT_FLAG
Access	R/W	R	R/W	R
Default	0	-	0	-

Bit 3: OVERRUN_INT_ENA – Configures interrupt for OVERRUN_INT_FLAG flag
 0 = Disables generation of interrupt for OVERRUN_INT_FLAG, Bit 2 (Default)
 1 = Enables interrupt generation for OVERRUN_INT_FLAG

Bit 2: OVERRUN_INT_FLAG – Over Run Flag
 0 = No overrun condition observed
 1 = Overrun condition detected

Bit 1: SPI_INT_ENA – Enables interrupt for SPI_INT_FLAG
 0 = Disables interrupt upon receipt of SPI_INT_FLAG (Default)
 1 = Enables generation of interrupt upon receipt of SPI_INT_FLAG

Bit 0: SPI_INT_FLAG – This bit is a read clear bit. It is set when an interrupt is generated, that is, the SPI has finished its transmission. During normal operation, a read of the shift Register implicitly clears this flag bit. A write of 0 to SPI_INT_FLAG explicitly clear the flag. During emulation mode, however, a read to the shift Register does not clear this flag bit. A write of 0 to SPI_INT_FLAG during emulation mode clears the flag. If the SPI_INT_ENA bit is 0, then this bit is cleared.

3.4 SPI Shift Register 0 (SPIDAT0)

Address FFF7F80C

Bit Number	31:0
Bit Name	SPIDAT
Access	R/W
Default	0000_0000_0000_0000_0000_0000_0000_0000

Bits 31-0: SPIDAT – These bits make up the SPI shift Register. Data is shifted out of the MSB (bit 31) and into the LSB (bit 0). SPIEN must be set to 1 before this Register can be written to. Writing a '0' to the SPIEN Register forces the 32 bits of the Register to be 0.

3.5 SPI Shift Register 1 (SPIDAT1)

Address FFF7F810

Bit Number	31:0
Bit Name	SPIDAT
Access	R/W
Default	0000_0000_0000_0000_0000_0000_0000_0000

Bits 31-0: SPIDAT – These bits make up the SPI Shift Register. Data is shifted out of the MSB (bit 31) and into the LSB (bit 0). SPIEN must be set to 1 before this Register can be written to. Writing a '0' to the SPIEN Register forces the 32 bits of the Register to be 0.

Formatted: Bullets and Numbering

3.6 SPI Receive Buffer Register (SPIBUF)

Address FFF7F814

Bit Number	31:0
Bit Name	SPIDAT
Access	R
Default	0000_0000_0000_0000_0000_0000_0000_0000

Bits 31-0: SPIDAT – These bits make up the SPI Receive Buffer Register. Data is shifted in from LSB (bit 0) to MSB (bit 31).

3.7 SPI Pin Control Register 1 (SPIPC1)

Address FFF7F81C

Bit Number	4	3	2	1	0
Bit Name	SCS_DIR	SOMI_DIR	SIMO_DIR	CLK_DIR	RESERVED
Access	R/W	R/W	R/W	R/W	-
Default	0	0	0	0	0

Bit 4: SCS_DIR – Configures direction of SPI_CS pin

0 = SPI_CS configured as an input (Default)

1 = SPI_CS configured as an output

Bit 3: SOMI_DIR – Configures direction of SPI_SOMI pin

0 = SPI_SOMI configured as an input (Default)

1 = SPI_SOMI configured as an output

Bit 2: SIMO_DIR – Configures direction of SPI_SIMO pin

0 = SPI_SIMO configured as an input (Default)

1 = SPI_SIMO configured as an output

Bit 1: CLK_DIR – Configures direction of SPI_CLK pin

0 = SPI_CLK configured as an input (Default)

1 = SPI_CLK configured as an output

Bit 0: RESERVED

Deleted: <#>¶

Formatted: Indent: Before: 0.13", Don't keep lines together, Tabs: 1", List tab + Not at 0.4" + 0.88"

Formatted: Bullets and Numbering

Formatted: No widow/orphan control

Formatted: No widow/orphan control, Don't keep lines together

Formatted: No widow/orphan control, Don't keep lines together

Formatted: No widow/orphan control, Don't keep lines together

Formatted: No widow/orphan control, Don't keep lines together

Formatted: No widow/orphan control, Don't keep lines together

Formatted: No widow/orphan control, Don't keep with next, Don't keep lines together

Formatted: Indent: Before: 0", No widow/orphan control

Formatted: Indent: Before: 0", No widow/orphan control

3.8 SPI Pin Status Register 1 (SPIST1)

Address FFF7F820

Bit Number	4	3	2	1	0
Bit Name	SCS_DIN	SOMI_DIN	SIMO_DIN	CLK_DIN	STE_DIN
Access	R	R	R	R	R
Default	-	-	-	-	-

Bit 4: SCS_DIN – Input value of SPI_CS pin

- 0 = Logic level low observed on SPI_CS pin
- 1 = Logic level high observed on SPI_CS pin

Bit 3: SOMI_DIN – Input value of SPI_SOMI pin

- 0 = Logic level low observed on SPI_SOMI pin
- 1 = Logic level high observed on SPI_SOMI pin

Bit 2: SIMO_DIN – Input value of SPI_SIMO pin

- 0 = Logic level low observed on SPI_SIMO pin
- 1 = Logic level high observed on SPI_SIMO pin

Bit 1: CLK_DIN – Input value of SPI_CLK pin

- 0 = Logic level low observed on SPI_CLK pin
- 1 = Logic level high observed on SPI_CLK pin

Bit 0: STE_DIN – Input value of ENABLE pin

- 0 = Logic level low observed on ENABLE pin
- 1 = Logic level high observed on ENABLE pin

Deleted: ¶

Formatted: Bullets and Numbering

3.9 SPI Pin Control Register 3 (SPIPC3)

Address FFF7F824

Bit Number	4	3	2	1	0
Bit Name	SCS_DOUT	SOMI_DOUT	SIMO_DOUT	CLK_DOUT	STE_DOUT
Access	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0

Bit 4: SCS_DOUT – Output value of SPI_CS pin

- 0 = Logic level low driven on SPI_CS pin (Default)
- 1 = Logic level high driven on SPI_CS pin

Bit 3: SOMI_DOUT – Output value of SPI_SOMI pin

- 0 = Logic level low driven on SPI_SOMI pin (Default)
- 1 = Logic level high driven on SPI_SOMI pin

Bit 2: SIMO_DOUT – Output value of SPI_SIMO pin

- 0 = Logic level low driven on SPI_SIMO pin (Default)
- 1 = Logic level high driven on SPI_SIMO pin

Bit 1: CLK_DOUT – Output value of SPI_CLK pin

- 0 = Logic level low driven on SPI_CLK pin (Default)
- 1 = Logic level high driven on SPI_CLK

Bit 0: STE_DOUT – Output value of ENABLE pin

- 0 = Logic level low driven on ENABLE pin (Default)
- 1 = Logic level high driven on ENABLE pin

Formatted: Bullets and Numbering

3.10 SPI Pin Control Register6 (SPIPC6)

Formatted: Bullets and Numbering

Address FFF7F830

Bit Number	4	3	2	1	0
Bit Name	SCS_FUN	SOMI_FUN	SIMO_FUN	CLK_FUN	STE_FUN
Access	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0

Bit 4: SCSFUN – SPI_CS Configuration

0 = Configured in GPIO mode (Default)

1 = Configured in SPI mode

Bit 3: SOMIFUN – SPI_SOMI Configuration

0 = Configured in GPIO mode (Default)

1 = Configured in SPI mode

Bit 2: SIMOFUN – SPI_SIMO Configuration

0 = Configured in GPIO mode (Default)

1 = Configured in SPI mode

Bit 1: CLKFUN – SPI_CLK Configuration

0 = Configured in GPIO mode (Default)

1 = Configured in SPI mode

Bit 0: ENABLEFUN – ENABLE configuration

0 = Configured in GPIO mode (Default)

1 = Configured in SPI mode

4 Useful C statements examples

```
SpilRegs.SPICTRL1.bit.CHAR_LEN = 15;  
// Means: SPI interface configured for data frame  
//transmission/reception length of 16 bits
```

```
SpilRegs.SPICTRL1.bit.PRESCALE = 1;  
// Means: SPI baud rate is set to 15.6 Mhz/2 = 7.8 Mbps
```

```
SpilRegs.SPICTRL2.all = 0x1C;  
// Means: SPI Clock is not phased, SPI Clock non-inverted, set to //  
Master Mode, SPI Enabled, SPI drives the clock pin
```

```
SpilRegs.SPICTRL3.bit.OVERRUN_INT_FLAG = 0;  
// Reading this bit has the same effect (Clear on read)  
// Means : Explicitly clear the SPI interrupt flag. Should be cleared  
// in the relevant ISR.
```

```
SpilRegs.SPIDAT0.all = data_word;  
// Initiates the transmission of data_word variable in 3-pin  
//configuration.
```

```
SpilRegs.SPIDAT1.all = data_word;  
// Initiates the transmission in 4-pin configuration. Lowers the SPI  
//chip select pin to enable the slave's output
```

```
SpilRegs.SPIPC1.all = SPIPC1_ALL_SCS_DIR | SPIPC1_ALL_SIMO_DIR |  
SPIPC1_ALL_CLK_DIR;  
// Means: SPICS, SPISIMO and SPICLK pins are configured as outputs,  
//SPISOMI is configured as input
```