2 Overview of the DW1000

2.1 Introduction

The DW1000 consists of an analog front-end (both RF and baseband) containing a receiver and transmitter and a digital back-end that interfaces to a host processor, controls the analog front-end, accepts data from the host processor for transmission and provides received data to the host processor over an industry standard SPI interface. A variety of control schemes are implemented to maintain and optimize transceiver performance.

2.2 Interfacing to the DW1000

2.2.1 The SPI Interface

The DW1000 host communications interface is a slave-only Serial Peripheral Interface (SPI) compliant with the industry protocol. The host system must include a master SPI bus controller in order to communicate with the DW1000.The SPI bus signals, their voltage levels and signal timings are described in the DW1000 data sheet.

The host system reads and writes DW1000 registers via the SPI. This section describes the format of the SPI transactions. For details of the SPI physical circuits, operational mode configuration and timing parameters please refer to the DW1000 data sheet. The SPI-accessible registers of the DW1000 are detailed in section7*– The DW1000 register set*.

2.2.1.1 SPI operating modes

The operating mode of the SPI is determined when the DW1000's digital control function is initialised as a result of a device reset or as it is woken up from a sleep state. At this time GPIO lines 5 and 6 are sampled and their values act to select the SPI mode.

It is possible to set the SPI mode within the DW1000's one-time programmable configuration block to avoid needing any external components and leave the GPIO free for alternative use. This is a one-time activity and cannot be reversed so care must be taken to ensure that the desired SPI mode is set. Please refer to section*6.3–Using the on-chip OTP* memoryfor more details of OTP configuration, and *Register file: 0x2D – OTP Memory Interface*.

For full details of the SPI operating modes and their configuration, please refer to the DW1000 data sheet.

2.2.1.2 Transaction formats of the SPI interface

Each SPI transaction starts with a one to three octet transaction header followed by a variable number of octets making up the transaction data. The number of data bytes allowed in an SPI transfer is not limited.

The transaction header selects whether the transaction is a read or a write and specifies the address to read from or write to. Physically the SPI interface is full duplex in that every transaction shifts bits both into and out of the DW1000. Logically however each transaction is either reading data from the DW1000 or writing data to it. As shown in Figure 1, for a read transaction all octets beyond the transaction header are ignored by the DW1000, and for a write transaction all octets output by the DW1000 should be ignored by the host system.

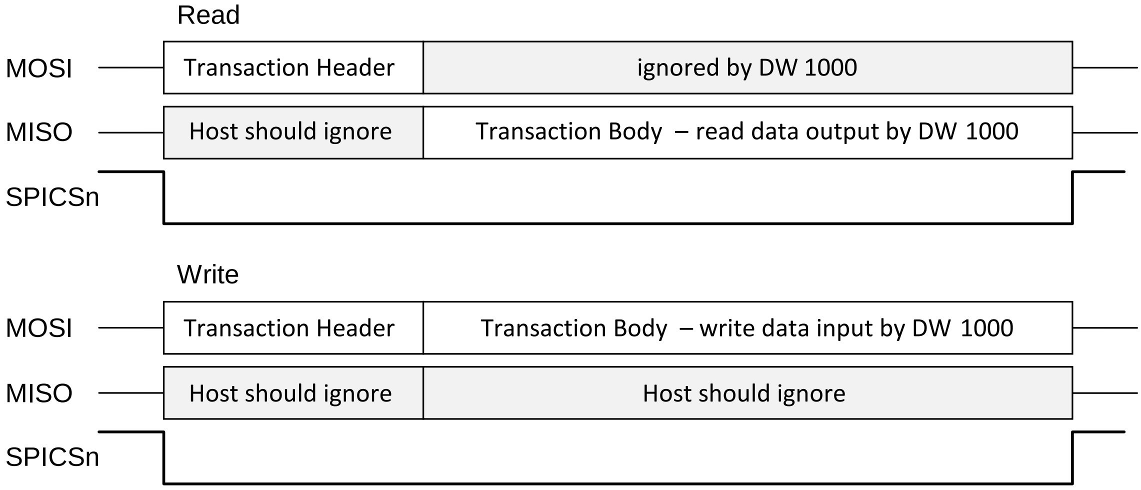


Figure 1: SPI Read and Write Transactions

Note: The octets are physically presented on the SPI interface data lines with the high order bit sent first in time.

SPI transactions are enveloped by the assertion of the active low chip select line, SPICSn. The high-to-low assertion (low) of SPICSn initialises the SPI transaction handler so that the DW1000 interprets the next octet(s) as a new transaction header. The low-to-high de-assertion of SPICSn ends the SPI transaction. Typically a software SPI interface driver will include a parameter to indicate the length of the transaction, i.e. how many octets to write to the device on the SPI bus, or how many bytes to read.

The SPI accessible parameters of the DW1000 are organised into 64 separate register file locations (detailed in section7 *– The DW1000 register set*). Every SPI access transaction header includes a 6-bit register file ID that identifies which register file is being accessed by the transaction. Sub-addressing within the selected register file allows efficient access to all the parameters within the DW1000. Depending on the sub-addressing being used, the transaction header is either one, two or three octets long. These three types of transaction are described in the sub-sections below.

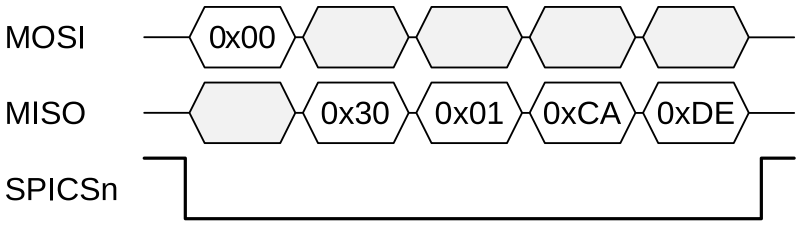
Note: when writing to any of the DW1000 registers care must be taken not to write extra data beyond the published length of the selected register (see section 7 *– The DW1000 register set*).

Figure 2 shows the fields within the one octet transaction header of a simple non-indexed SPI transaction. Bit-6 is zero indicating that a sub-index is not present. The register (file) ID selects the top level addressing of the DW1000 parameter or parameter block being accessed.

Bit number:	7	6	5	4	3	2	1	0	
Meaning:	Operation: 0 = Read 1 = Write	Bit = 0, says sub-index is not present	Register file ID – Range 0x00 to 0x3F (64 locations)						Transaction Header Octet

Figure 2: Single octet header of the non-indexed SPI transaction

The remaining octets of the transaction, the transaction body, immediately following this one-octet header are read from (or written to) the selected register file beginning at index zero. Figure 3 shows an example of a non-indexed read from the Device ID register using the single octet header.



Non sub indexed read from register file ID 0x00, this 32-bit register contains the Device ID, the 4-octet result is 0xDECA0130

Figure 3: Example non-indexed read of the Device ID register (0x00)

Note: The octets of a multi-octet value are transferred on the SPI interface in octet order beginning with the low-order octet. This is shown in Figure 3.

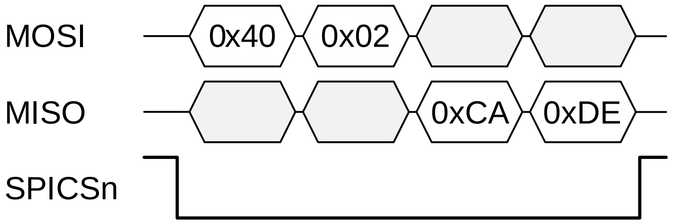
2.2.1.2.1 SPI transaction with a 2-octet header

Figure 4 shows the fields within the two octet transaction header of a short-indexed SPI transaction. Bit-6 of the first octet is 1 indicating that a sub-index is present. The register (file) ID in the first octet selects the top level address of the DW1000 parameter block being accessed. In the second octet bit-7 is zero indicating that a further transaction header octet is not present and that the remaining 7 bits of octet-2 are a short sub-index into the register file.

Bit number:	7	6	5	4	3	2	1	0	
Meaning:	Operation: 0 = Read 1 = Write	Bit = 1, says sub-index is present	Register file ID – Range 0x00 to 0x3F (64 locations)						Transaction Header Octet 1
	Extended Address: 0 = no	7-bit Register File sub-address, range 0x00 to 0x7F (128 byte locations)							Octet 2

Figure 4: Two octet header of the short indexed SPI transaction

The remaining octets of the transaction, the transaction body, immediately following this two-octet header are read from (or written to) the selected register file beginning at the selected index address 0 to 127. Figure 5 shows an example of an indexed read from the Device ID register using the two octet transaction header.



Short sub-indexed read 2-octets beginning at index 2 in register file ID 0x00. This reads the high two octets of this 32-bit register, the result is 0xDECA.

Figure 5: Example short-indexed read of 3rd and 4th octets of register 0x00

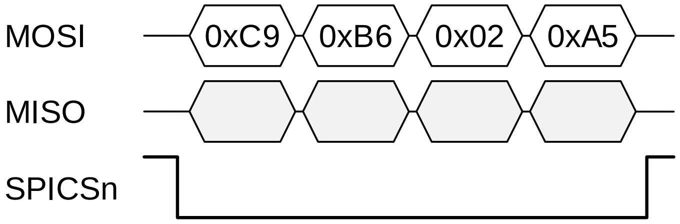
2.2.1.2.2 SPI transaction with a 3-octet header

Figure 6 shows the fields within the three octet transaction header of a long-indexed SPI transaction. Bit-6 of the first octet is 1 indicating that a sub-index is present. The register (file) ID in the first octet selects the top level addressing of the DW1000 parameter or parameter block being accessed. In the second transaction header octet bit-7 is set indicating the long form of indexed addressing is to be employed and thus the remaining seven bits of the second octet along with all of the third transaction header octet form a 15-bit sub-index into the selected register file.

Bit number:	7	6	5	4	3	2	1	0	
Meaning:	Operation: 0 = Read 1 = Write	Bit = 1, says sub-index is present	Register file ID – Range 0x00 to 0x3F (64 locations)						Transaction Header Octet 1
	Extended Address: 1 = yes	Low order 7 bits of 15-bit Register file sub-address range 0x0000 to 0x7FFF (32768 byte locations)							Octet 2
	High order 8 bits of 15-bit Register file sub-address range 0x0000 to 0x7FFF (32768 byte locations)								Octet 3

Figure 6: Three octet header of the long indexed SPI transaction

The octets of transaction body which immediately follow the transaction header are read from (or written to) the selected register file beginning at the selected sub-index address 0 to 32767.



Long sub indexed write of octet value 0xA5 to the transmit buffer (Reg ID 0x09) at index 310 (0x136 in hex)

Figure 7: Example long-indexed write of one octet to index 310 of the TX buffer

Figure 7 shows an example of an indexed write that uses the longer the three octet header. This example is a write to the transmit data buffer at sub index 0x136. The TX buffer has register file ID of 0x09. Octet 1 of transaction header is thus 0xC9 as bit-7 is 1 to signal a write and bit-6 is 1 indicating a sub-address follows. The 15-bit sub-address has the binary value 000-0001-0011-0110. In octet 2 of the transaction header, bit 7 is set to indicate an extended sub-index and the remaining bits contain 0110110, the low 7 bits of the sub-address. Octet 3 of the transaction header then contains 00000010, the remaining eight high order bits of the sub-address index, which is 0x02 in hex.

The DW1000 parameters that may be read and written using these SPI transactions are detailed in section7 *– The DW1000 register set*.

2.2.2 Interrupts

The DW1000 can be configured to assert its IRQ pin on the occurrence of one or more status events. The assertion of the IRQ pin can be used to interrupt the host controller and redirect program flow to deal with the cause of the event.

The polarity of the IRQ pin may be configured via the HIRQ_POL bit in the *Register file: 0x0D – System Control Register* .By default on power up the IRQ polarity is active high.

The occurrence of a status event in *Register file: 0x0F – System Event Status Register* may assert the IRQ pin depending on the setting of the corresponding bit in the *Register file: 0x0E – System Event Mask Register.*

By default, on power-up, all interrupt generating events are masked and interrupts are disabled.

2.2.3 General Purpose I/O

The DW1000 provides 8 GPIO pins. These can be individually configured at the user’s discretion to be inputs or outputs. The state of any GPIO configured as an input can be read and reported to the host controller over the SPI interface. When configured as an output the host controller can set the state of the GPIO to high or low.

Some of the GPIO lines have multiple functions as listed in the DW1000 data sheet.

The configuration and operation of the GPIO pins is controlled via *Register file: 0x26 – GPIO control and status*

By default, on power-up, all GPIOs are configured as inputs.

2.2.4 The SYNC pin

This pin is used for external clock synchronisation purposes. See section *6.1 – External Synchronisation* for further details.

2.3 DW1000 Operational States

2.3.1 State diagram

The DW1000 has a number of different operational states (or modes). These are listed and described in Table 1below and the transitions between them are illustrated in Figure 8.