

SPI Slave Mode

The synchronous serial interface comprises the SPI_CLK (input) SPI_MOSI (output), SPI_MISO (input), SPI_CS (input) and SPI_INT (output) signals. When used in SPI mode, these signals can be read (but not written) as GPIO port bits.

Using this interface, an external microcontroller can be provided with a network connection it would otherwise lack, using the module as a modem. The microcontroller can set up and tear down TCP/IP connections as a client or as a server, and send and receive data packets.

A simple protocol is used for communicating over the SPI bus. The module is viewed as containing a set of registers (see **Table 2**) that mediate the process. A request by the SPI bus master starts by activating the SPI_CS signal and sending a single control byte (see **Table 1**). The module replies with a one byte delay, and so the first byte received from the module must be discarded by the master.

The master can then either send a sequence of bytes, or zero in order to terminate the request. If the module replies with a zero byte, that means that it has received invalid data and the master must terminate the request with a zero: it must deassert SPI_CS and can then begin a new request. Any zero bytes to be communicated on the bus must be escaped using a backslash as described in the main article. Each byte received by the module increments its internal address counter, which allows a sequence of registers to be set up in a single command. (This does not apply to the buffer register at address 0x18.) In a read command, the master sends out (non-zero) dummy bytes to the module, while the module replies with the contents of the addressed registers. In a write command, the module acknowledges the bytes it receives with dummy bytes.

Also, whenever the status of the module changes, it asserts (that is, takes low) the SPI_INT signal. Reading the flag register (at address 0x01) will reveal the reason for the status change and deassert SPI_INT.

Module state on reset

Upon reset the registers in the module are all initialised to zero. The module starts up and, assuming nothing goes awry, asserts that it is *READY* (address 0x02). A network connection can only be set up from the *READY* state.

Using the module as a server

The *LISTEN* command activates the serial server in an analogous way to SPI master mode. Data packets can be exchanged once a connection has been successfully established with a client. The server remains active in this mode until the next reset.

Using the module as a client

In this case an IP address (starting at register 0x10) and port number (starting at address 0x14) must be set, and then a *CONNECT* command sent to the module. The module will then automatically handle ARP requests using this IP address. Again, data packets can be exchanged once a connection has been successfully established.

Data packet exchange

The data buffer register (at address 0x18) is different from the others in that it provides automatic sequential access to the receive buffer (when read) or transmit buffer (when

written). SPI_INT is asserted when a data packet is received, and the corresponding flag is set in the flag register. The received data should be retrieved from the buffer as soon as possible after the interrupt is signalled. Data written to the buffer are sent immediately: there is no explicit command to flush the buffer.

Bit 7

0: read access
1: write access

Bits 6 to 0

Register address at which access starts (see Table 2)

Table 1: Control byte (8 bits)

Register address	Name	Description
0x00: read only	Reserved	
0x01: read only	Flags	Set flags trigger an interrupt (SPI_INT=0); reading the register clears it and deasserts SPI_INT (see Table 3)
0x02: read only	Status	General module status (see Table 4)
0x03: read only	Software version	Major version number
0x04: read only	Software version	Minor version number
0x05: read only	Software version	Build number
0x06 – 0x09: read only	IP address	Current module IP address as four bytes
0x0A	Mailbox	Commands to module (see table 5)
0x0B to 0x0F	Reserved	
0x10 to 0x13	Peer IP address	IP address of connected device as four bytes (in client mode, must be set before connection is set up; in server mode, will be set by module after connection is set up)
0x14 to 0x15	Port	Port number, little-endian format (in client mode, port number on connected device; in server mode, port number on module)
0x16	Reserved	
0x17	Socket mailbox	When read, gives status information (see Table 7); when written, sends commands (see Table 6)
0x18	Data buffer	When read, returns bytes from receive buffer; when written, sends data

Table 2: Registers

Bits 7 to 3	Bit 2	Bit 1	Bit 0
Reserved	0: Receive buffer is empty 1: Characters available in receive buffer for reading from address 0x18	0: Socket status unchanged 1: Socket status has changed; new status can be read at address 0x17	0: Module status unchanged 1: Module status has changed; new status can be read at address 0x02

Table 3: Flags

Status	Name	Description
0x00	STARTING	Module in process of initialisation
0x01	NOLINK	Module has no network connection
0x02	IP_ERROR	Module has detected an IP address conflict

0x03	READY	Module has completed initialisation and is operating normally
0x04	SHUTDOWN	Module is about to shut down

Table 4: Module status codes

Command	Name	Description
0x00	NOP	Null command, ignored by module
0x01	RESTART	Resets module
0x02	SHUTDOWN	Shuts module down

Table 5: Module commands

Command	Name	Description
0x00	NOP	Null command, ignored by module
0x01	LISTEN	Starts up the serial server using either the port as specified by the register, or, if this is zero, the default port number as configured using the web interface
0x02	CONNECT	Attempts to establish a connection to the given IP address and port number
0x03	DISCONNECT	Tears down an established connection

Table 6: Socket commands

Command	Name	Description
0x00	IDLE	There is no connection and the serial server is not running
0x01	LISTENING	The serial server has been started and is listening on the configured port
0x02	CONNECTED	A connection has been established, and the IP address of the connected device is available from the registers starting at address 0x10
0x03	DISCONNECTING	The connection is in the process of being torn down
0x04	CONNECTING	The connection is in the process of being set up

Table 7: Socket status